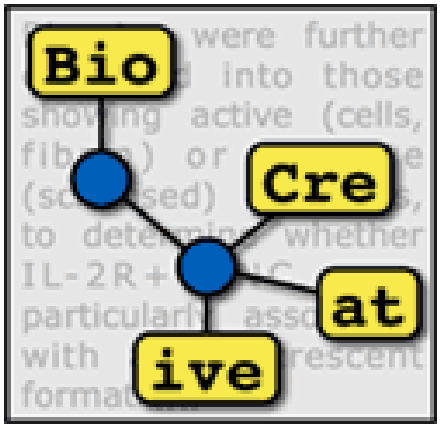
Proceedings of the BioCreative VI Workshop



October 18-20, 2017
Bethesda, MD USA

Editors:

Cecilia Arighi
Qinghua Wang
Cathy Wu

BioCreative VI Workshop Proceedings
Table of Contents

Preface…….3
Committees and Sponsors……4
Scientific Program……..5
Keynotes and Panels…….6-12
Track 1: Bio-ID Assignment……..13-39
Track 2: Text Mining Services for Kinome Curation…….40-55
Track 3: Extraction of causal network information using BEL…….56-81
Track 4: Mining protein interactions and mutations for
precision medicine …….82-139
Track 5: Text mining chemical-protein interactions …….140-193
General Session……194
Poster Session…….203

Preface

Welcome to the BioCreative VI workshop which is being held in Bethesda MD, USA on October 18-20, 2017. On behalf of the Organizing Committee, we would like to thank you for your participation and hope you enjoy the workshop.

The BioCreative (Critical Assessment of Information Extraction systems in Biology) challenge evaluation consists of a community-wide effort for evaluating text mining and information extraction systems applied to the biological domain (http://www.biocreative.org/). Its aim is to promote the development of text mining and text processing tools which are useful to the communities of researchers, publishers, and database curators in the biological sciences. The main emphasis is on the comparison of methods and the community assessment of scientific progress, rather than on the purely competitive aspects.

The first BioCreative was held in 2004, and since then each challenge has ensure the active involvement of the text mining user community in the design of the tracks, preparation of corpus and the testing of interactive systems. BioCreative VI consists of five tracks:

Track 1: Interactive Bio-ID Assignment for innovations in Biomedical Digital Curation. Organizers: Lynette Hirschman, Cecilia Arighi, Thomas Lemberger, Robin Liechti and Cathy Wu. The Bio-ID track explores the ID assignment to selected bioentities both at the pre- and post-publication , using the SourceData framework as a real world use case.

Track 2: Text-mining services for Kinome Curation. Organizers: Julien Gobeill, Patrick Ruch and Pascale Gaudet. This track focuses on literature triage (selection of relevant articles for curation), a basic task performed by virtually all curated molecular biology databases. More specifically on triage for both Protein-Disease and Protein-GO annotations related to human kinases.

Track 3: Extraction of causal network information using the Biological Expression Language (BEL). Organizers: Juliane Fluck, Sumit Madan and Justyna Szostak. BEL track focuses on automatic extraction of biological network information, one of the most desired and most complex tasks in biological and medical text mining, represented in Biological Expression Language (BEL).

Track 4: Mining protein interactions and mutations for precision medicine. Organizers: Rezarta Islamaj Dogan, Andrew Chatr-aryamontri, Sun Kim, Donald C. Comeau, Zhiyong Lu. This track aims to bring together the biomedical text mining community in a new challenge for precision medicine, focusing on identifying and extracting protein-protein interactions affected by mutations described in the biomedical literature.

Track 5: Text mining chemical-protein interactions. Organizers: Martin Krallinger, Alfonso Valencia, Analia Lourenço. This track focuses on chemical-protein interactions that might be of relevance for precision medicine as well as for drug discovery and basic biomedical research.

We would like to thank all participating teams, panelists, all the chairs, sponsors and committee members.

Organizing Chairs
Cecilia Arighi, University of Delaware, USA
Cathy Wu, University of Delaware and Georgetown University, USA

BioCreative VI Committees and Sponsors

Steering Committee

Cecilia Arighi, University of Delaware, USA
Andrew Chatr-aryamontri, IRIC, Université de Montréal, Canada
Donald Comeau, National Center for Biotechnology Information (NCBI), NIH, USA
Kevin Cohen, University of Colorado, USA
Juliane Fluck, Fraunhofer Institute for Algorithms and Scientific Computing SCAI, Germany
Sumit Madan, Fraunhofer Institute for Algorithms and Scientific Computing SCAI, Germany
Rezarta Islamaj Dogan, NCBI, NIH, USA
Pascale Gaudet, Swiss Bioinformatics Institute, Switzerland
Julien Gobeill, Swiss Bioinformatics Institute, Switzerland
Lynette Hirschman, MITRE Corporation, USA
Sun Kim, NCBI, NIH, USA
Martin Krallinger, Spanish National Cancer Centre, CNIO, Spain
Zhiyong Lu, NCBI, NIH, USA
Fabio Rinaldi, Swiss Bioinformatics Institute, Switzerland
Patrick Ruch, Swiss Bioinformatics Institute, Switzerland
Alfonso Valencia, Spanish National Cancer Centre, CNIO, Spain
Cathy Wu, University of Delaware and Georgetown University, USA

Local Organizing Committee

Cecilia Arighi, University of Delaware, USA
Donald Comeau, NCBI, NIH, USA
Sun Kim, NCBI, NIH, USA
Rezarta Islamaj Dogan, NCBI, NIH, USA
Zhiyong Lu, NCBI, NIH, USA
Qinghua Wang, University of Delaware, USA
Cathy Wu, University of Delaware and Georgetown University

Sponsors

NIH/MIGMS Conference grant 1R13GM109648-01A1
NIH/MIGMS Supplement 2R01GM08064

Scientific Program

Wednesday, October 18, 2017

08:00 - 12:00	Registration (EMC foyer)
08:00 - 09:00	Breakfast (EMC foyer)
09:00 - 09:15	Workshop opening
09:15 - 10:40	**TRACK 1 Interactive Bio-ID Assignment** 9:15-9:25 Introduction to Bio_ID track (Cecilia Arighi, University of Delaware) 9:25-9:40 Introduction to SourceData and data set (Thomas Lemberger, SourceData) 9:40-9:55 Overview of Bio_ID results – batch results (Lynette Hirschman, MITRE) 9:55-10:05 A Neural Named Entity Recognition Approach to Biological Entity Identification (Emily Sheng, Information Sciences Institute/USC) 10:05-10:15 A Study on Identification of Organism and micro-RNA Mentions in Figure Captions (Po-Ting Lai, National Central University) 10:15-10:40 Next steps and discussion
10:40 - 11:00	Break
11:00 - 12:30	**Panel on Innovation on Digital Curation** (Moderators: Fabio Rinaldi and Cecilia Arighi) 11:00-11:10 Julio Collado-Vides, UNAM- Natural language processing to enhance accessibility to knowledge in RegulonDB 11:10-11:20 Thomas Lemberger, EMBO- Data transparency in scientific publishing 11:20-11:30 Zhiyong Lu, NCBI- Text mining for improving the prioritization, curation, and integration of knowledge for clinically relevant variants 11:30-11:40 Johanna McEntyre, EMBL-EBI- How can text mining scale to meet diverse and precise curation needs? 11:40-12:30 Open discussion
12:30 - 13:40	Lunch (Restaurant first floor)
13:40 - 15:40	**TRACK 4 Mining protein interactions and mutations for precision medicine** 1:40-2:10 Overview of the Precision Medicine Track (Rezarta Islamaj Dogan) 2:10-2:30 Identifying Relevant Literature for Precision Medicine Using Deep Neural Networks (Sergio Matos) 2:30- 2:50 Exploring a Deep Learning Pipeline for the BioCreative VI Precision Medicine Task (Tung Tran) 2:50- 3:15 Mining protein interactions affected by mutations using a NLP based machine learning approach (Albert Steppi and Jinchan Qu) 3:15-3:30 Document Triage and Relation Extraction for Protein-Protein Interactions affected by Mutations (Dina Demner Fushman for Karin Verspoor and team) 3:30-3:40 Poster spotlight and Open discussion
15:40 - 16:00	Break
16:00 - 17:00	**Keynote- Dr. Patricia Flatley Brennan, Director of National Library of Medicine, NIH** Towards a future of data-powered health
17:00 - 18:00	**Panel on Funding Stakeholders** (Moderator: Cathy Wu) Susan Gregurick, NIGMS, NIH Jennifer Weller, NSF Jane Ye, NLM, NIH

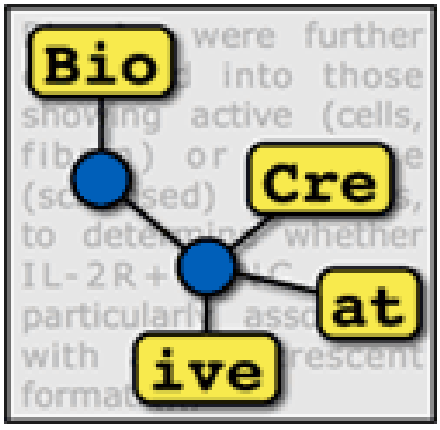
Thursday, October 19, 2017

08:00 - 12:00	Registration (EMC foyer)
08:00 - 09:00	Breakfast (EMC foyer)
09:00 - 10:15	**General session** 9:00-9:20 Efficient and Accurate Entity Recognition for Biomedical Text (Fabio Rinaldi, U. Zurich) 9:20-9:40 iTextMine: Integrated Text Mining System for Large-Scale Knowledge Extraction from Literature (Jia Ren, U. Delaware) 9:40-10:00 Large-scale Automated Reading with Reach Discovers New Cancer Driving Mechanisms (Dane Bell, U. Arizona) 10:00-10:20 Evaluating without a Gold Standard (Lynette Hirschman, MITRE)
10:20 - 10:40	Break
10:40 - 12:30	**TRACK 2 Text-mining services for Kinome Curation** 10:40-11:10 Kinome Track Overview (Julien Gobeill & Patrick Ruch, SIB) 11:10-11:35 Assisting Document Triage for Human Kinome Curation via Machine Learning (Alan Hsu, NCBI) 11:35-11:55 KinDER: A Biocuration Tool for Extracting Kinase Knowledge from Biomedical Literature (Adam Morrone & Daniel Dopp, Montana State University) 11:55-12:20 Discussion
12:30 - 13:45	Lunch (Restaurant first floor)
13:45 - 15:45	**TRACK 3 Extraction of causal network information using the Biological Expression Language** Overview and Results (Sumit Madan, Fraunhofer SCAI) Selected Team Presentations Next steps and discussion (Sumit Madan, Fraunhofer SCAI)
15:45 - 16:00	Break
16:00 - 17:00	**Keynote- Dr. Hongfang Liu, Professor Biomedical Informatics, Mayo Clinic** Text mining in precision medicine: opportunities and challenges
17:00 - 19:00	**Poster session and Reception**

Friday, October 20, 2017

08:00 - 09:00	Breakfast (EMC foyer)
08:30 - 10:30	**TRACK 5 Text mining chemical-protein interactions** 8:30-9:00 Overview of the Chemical-Protein relation extraction track (Martin Krallinger / Saber A. Akhondi (CNIO / Elsevier) 9:00-9:15 Chemical-protein relation extraction with SVM, CNN, RNN and ensemble systems (Yifan Peng, NCBI, NLM, NIH) 9:15-9:30 Extracting Chemical-Protein Interactions using Long Short-Term Memory Networks (Sérgio Matos, University of Aveiro, Portugal) 9:30-9:45 TAttention based Neural Networks for Chemical Protein Relation Extraction (Ravikumar Komandur Elayavilli, Mayo Clinic, USA) 9:45-10:00 Extracting protein-chemical compound interactions from literature (Pei-Yau Lung, Florida State University) 10:00-10:15 Knowledge-base-enriched relation extraction (University of Colorado, Boulder) 10:15 - 10:30 CTCPI - Convolution Tree Kernel-based Chemical-Protein interaction detection (Po-Ting Lai, National Tsing-Hua University, Hsinchu, Taiwan)
10:30 - 10:45	Break
10:45 - 12:00	**Tracks open discussion/closing**
12:00 - 13:00	Lunch (Restaurant first floor)
13:00 - 15:30	**Tour to the National Library of Medicine**

Keynotes and Panels



Keynote 1: Towards a future of data-powered health

Patricia Flatley Brennan, RN, PhD
Director
National Library of Medicine

The National Library of Medicine, charged by the NIH to become the hub of data science, will launch its new strategic plan later this fall. NLM will accelerate its investments in advanced data management and stimulate new development in data science methodologies. In addition, NLM will explore new ways to reach out to its stakeholders in the scientific and clinical community, and place new emphasis on direct-to-patient resources that will bring the power of data science in to the everyday health lives of people. Expansion of our university-based training programs in medical informatics with new data science workforce development programs will foster the development of a scientific, clinical and library science workforce capable of leveraging data-driven science into new discoveries for health. Key to our future is embracing the tenets and practices under the open science umbrella, creating an infrastructure that supports data sharing, provides incentives for participation, and monitors the use and impact of these strategies in achieving health for all.

Keynote 2: Text mining in precision medicine: opportunities and challenges

Hongfang Liu, PhD
Professor Biomedical Informatics
Mayo Clinic

The promise of precision medicine is to enable medical practitioners to make better clinical decisions through incorporating individual variability in genes, environment, and lifestyle for each person. The anticipation of precision medicine is to increase prevention and prediction of disease, enable earlier disease intervention, improve targeted therapies, reduce side effects, and therefore reduce healthcare costs with better patient outcomes. A huge amount of information related to precision medicine is buried in text. Text mining techniques such as information retrieval or natural language processing (NLP) play a vital role in advancing precision medicine research and delivery. In this talk, we will first review literature and various activities in the text mining community for precision medicine. We will then introduce the precision medicine activities at Mayo Clinic. Some of our ongoing text mining efforts in contributing to the text mining community and enabling the use of text mining techniques for advancing precision medicine research and delivery institutionally will be discussed. Challenges faced in delivering text mining solutions for precision medicine will be discussed.

Panel: Innovation on Digital Curation

Natural language processing to enhance accessibility to knowledge in RegulonDB

Julio Collado-Vides**,** Center for Genomic Sciences, UNAM, Mexico

Abstract

The ultimate goal of curated databases is to facilitate access to data, information and knowledge. Data and information are usually modeled as objects or tables as part of the structured component of a database. Knowledge can be practically defined as all the additional statements in published journals that because of different reasons (limitations in the database modeling, as well as the rich expression of correlations, and interpretations in written language) are present in original publications but not in the structured items in databases because they are hard to curate and encode. I will present progress on how we have used natural language processing (NLP) to address knowledge encoding in RegulonDB where we have gathered for more than 20 years information on transcriptional regulation and operon organization in *E.coli* K12. We are working on NLP implementations both at the curation front as well as at the users front.

Acknowledgement: This project is supported by NIH-NIGMS grant number RO1GM110597

Panel: Innovation on Digital Curation

Data transparency in scientific publishing

Thomas Lemberger**,** EMBO

Abstract

Scientific progress depends on efficient sharing of reliable research findings. This implies transparency and openness, which make the scientific process traceable and accelerate the dissemination of research and thus the pace of scientific discovery. But is also requires critical evaluation and quality control steps to assess whether new claims are supported by evidence and represent true novel discoveries.

There are inherent tradeoffs between the need for immediacy and openness, the necessity of a scalable quality control process and the essential role of trust based on the responsible conduct of research by scientists.

In the life sciences, scientific journals disseminate peer-reviewed research on a large scale (more than 1 million articles per year). Moving scientific publishing towards more transparency raises important questions. How can journals add transparency to the peer-review and the editorial selection process? How can data integrity verifications be conducted in a scalable way? How will new technologies help to make published data and methods more discoverable and reproducible?

The rise of large-scale data mining and artificial intelligence will undoubtedly open new avenues in the way science is shared and published. Editors may transform into 'knowledge architects' while journals will become interoperable platforms that are readable both by machines and humans. As we may transition towards a quasi-immediate open sharing of data, it is also crucial to remember that costly human expertise, creativity and time-consuming critical scrutiny irreplaceably remain at the heart of the scientific enterprise.

Panel: Innovation on Digital Curation

Text Mining for Improving the Prioritization, Curation, and Integration of Knowledge for Clinically Relevant Variants

Zhiyong Lu**,** NCBI, NLM, NIH

Abstract

Understanding the associations of genomic variants with diseases and conditions and assessing their clinical significance is critical for genomic research precision medicine. Despite significant efforts in expert curation, information about most of the 154 million dbSNP Build 149 reference variants (RS) remains unknown. On the contrary, a wealth of human knowledge about the variant biological function and disease impact is buried in unstructured literature data. Previous studies have attempted to harvest and unlock such information with text-mining techniques but are of limited use in practice.

I will first present a new text-mining method (tmVar 2.0) for extracting variant mentions in the literature and subsequently normalizing them to standardized database identifiers, followed by a large-scale analysis of text-mined results vs. curated data from existing databases [1].

Next, through several real-world use cases (e.g. [2, 3]), I will demonstrate that our approach can identify high impact variants from publications and that our results can be combined with existing data to prioritize and rank the variants by various attributes (e.g. functional consequence and allele frequency). I will conclude by summarizing the opportunities and challenges of using text mining for the manual curation and interpretation of variation effects on biological functions and diseases to enrich our current knowledge.

References:

[1]: Wei C-H, Phan L, Feltz J, Maiti R, Hefferon T, Lu Z. tmVar 2.0: Integrating genomic variant information from literature with dbSNP and ClinVar for precision medicine. Bioinformatics (Oxford, England). btx541. https://doi.org/10.1093/bioinformatics/btx541

[2]: Poux S, Arighi CN, Magrane M, Bateman A, Wei CH, Lu Z, Boutet E, Bye-A-Jee H, Famiglietti ML, Roechert B, UniProt Consortium. On expert curation and scalability: UniProtKB/Swiss-Prot as a case study. Bioinformatics. 2017 Jul 13:btx439.

[3]: Singhal A, Simmons M, Lu Z. Text Mining Genotype-Phenotype Relationships from Biomedical Literature for Database Curation and Precision Medicine. PLoS computational biology. 2016;12(11):e1005017. doi:10.1371/journal.pcbi.1005017.

Panel: Innovation on Digital Curation

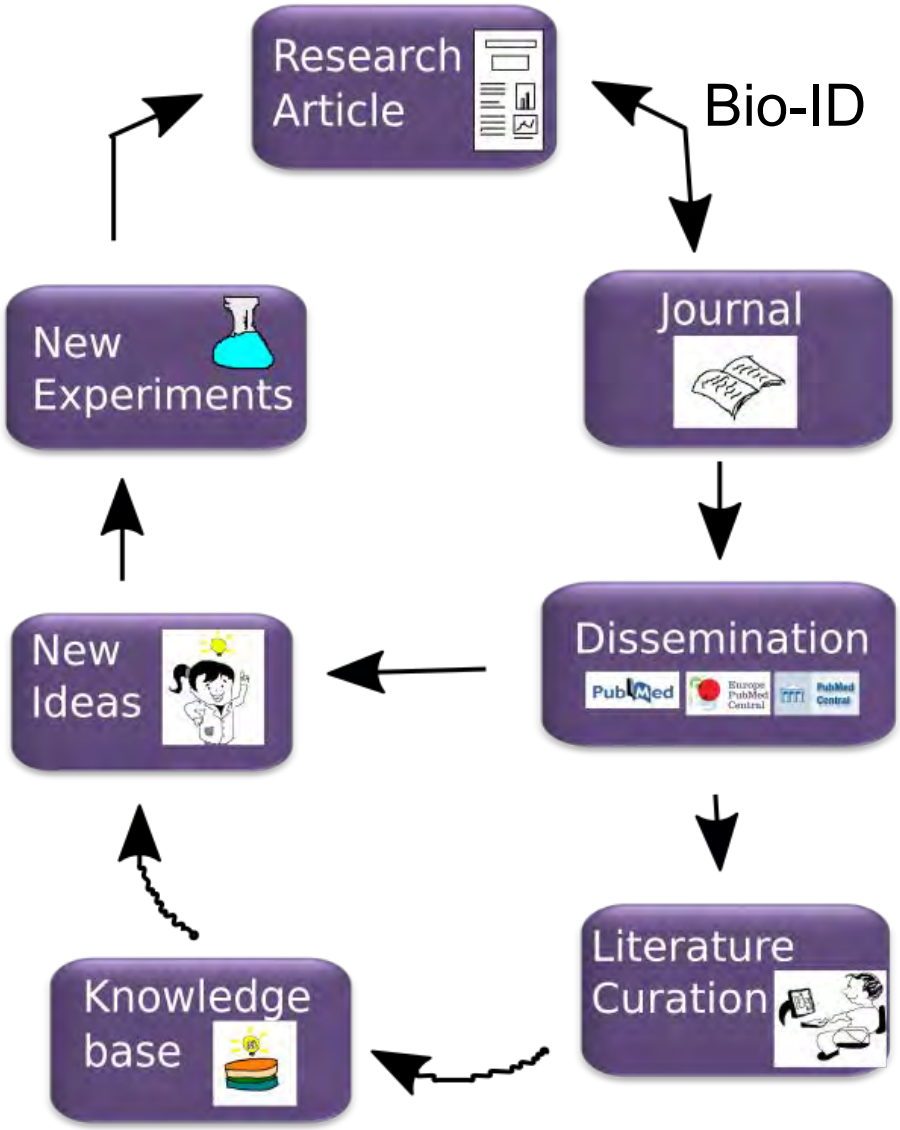
How can text mining scale to meet diverse and precise curation needs?

Jo McEntyre, EMBL-EBI

Abstract

Expert curation is critical to data quality and reuse in the life sciences. Without question, curation accelerates access to research results, synthesizing data from thousands of labs into records that can be digested straightforwardly by humans and machines alike. Typically, curation involves filtering through masses of potentially useful research articles, identifying the most critical ones and then extracting key information, usually placing this into the context of a knowledgebase. We are interested in exploring how text mining can support such curation efforts, and furthermore, how we can take a more infrastructural approach to this challenge. The typical working relationship between a group of curators with specific interests and a text mining group that strives to support those precise needs will not scale as we strive to meet the increasing demand for quality curation. Furthermore, a siloed way of working means that we may miss collaborative curation opportunities. At Europe PMC, we are exploring how to combine text-mining, open access content, indicators of scientific quality and curator feedback on the Europe PMC platform. The goal is to engage the text mining community and curators to contribute open annotations, which, in the context of the rich metadata available in Europe PMC, may support article prioritization systems, or scan-reading tools for ascertaining the quality of individual papers (e.g. via the SciLite application). A new APIs for retrieving all annotations will shortly be released to encourage distribution and reuse in other contexts. Feedback on text-mined entities and relationships, with a view to improving algorithms is also desirable, from curators or other knowledgeable readers. Finally, to improve integration between the literature and data resources, we have developed a deep linking mechanism that enables precise linking between curated articles and data resources.

Track 1
Bio-ID Assignment



Bio-ID Track Overview

Cecilia Arighi[1], Lynette Hirschman[2], Thomas Lemberger[3], Samuel Bayer[2], Robin Liechti[4], Donald Comeau[5], Cathy Wu[1]

[1]Dept. of Computer and Information Sciences, University of Delaware, Newark, DE, USA
[2]Information Technology Center, The MITRE Corporation, Bedford, M, USA
[3]EMBO, Heidelberg, Germany
[4]SIB, Lausanne, Switzerland
[5]NCBI, NIH, Bethesda, MD, USA

Abstract— **The Bio-ID track focuses on entity tagging and ID assignment to selected bioentity types, with the aim of facilitating downstream article curation both at the pre- and post-publication stages. The task is to annotate text from figure legends with the entity types and IDs for taxon (organism), gene, protein, miRNA, small molecules, cellular components, cell types and cell lines, tissues and organs. The track draws on SourceData annotated figure legends (by panel), in BioC format, and the corresponding full text articles (also BioC format) provided for context. Six teams submitted annotations: two teams submitted results for all 6 entity types; two teams submitted results for organism only; one team submitted results for miRNA and organism; and one team submitted results for small molecules. Mention level F-measures were 0.8 or better for cell type, species and gene-or-protein; however, micro-averaged normalized F-measure was significantly lower: 0.76 for species, 0.65 for cell type and below 0.6 for the other entity types. A subsequent experiment will investigate whether such systems can speed expert curation.**

Keywords—bioentity extraction; normalization; figure captions

I. Introduction

Innovations in biomedical digital curation have emerged as a critical topic to address sustainability of biological databases and research resources. Digital curation is defined as "the active management and preservation of digital resources over the lifecycle of scholarly and scientific interest, and over time for current and future generations of users" (1). In particular, there is a recognition that data curation needs to be integrated throughout the research lifecycle, without having to wait for curation by biocurators until after publication, as is the current practice for curated databases. While capturing knowledge of researchers at the time of data generation and publishing may enhance efficiency, there are significant barriers to moving curation "upstream." It is well recognized that the adoption of common database identifiers (IDs), controlled vocabularies (CVs) and ontologies facilitates data integration and re-use; however, it is nontrivial to extract IDs, CVs and ontological terms from the free texts of the scientific literature. New methods and tools need to be developed to support more effective and consistent curation at the time of paper submission.

The Bio-ID track addresses these needs for Innovations in Biomedical Digital Curation (2). Publications are one of the main vehicles for dissemination of experimental results. Researchers have new ideas, conduct experiments, write up their results summarizing those experiments, submit them to a journal and, if accepted after peer-review, the articles are disseminated in public literature databases. Publications are also the primary source of data for knowledgebase curators, who extract and summarize the relevant data in standard formats. While researchers use both the literature and knowledge bases, the latter offer efficient platforms for querying, given the linkage of data in literature to database objects. Then new ideas/hypotheses are generated to start a new cycle. Currently, there is a bottleneck in data re-use as curators spend time identifying bioentities in publications and linking these entities into their databases. We hypothesize that curation would be facilitated if articles were preprocessed to link the key bioentities to their appropriate biological knowledge bases, prior to publication (benefitting publishers) and prior to curation (speeding the downstream curation process); we refer to this as bio-ID assignment.

II. The Bio-ID Track

The Bio-ID track focuses on entity tagging and ID assignment to selected bioentity types, with the aim of facilitating downstream article curation at both the pre- and post-publication stages. This track builds on the SourceData project (3) as well as previous BioCreative experiments, including the interactive tracks (IAT), earlier gene/protein/chemical extraction tracks, the BioC interchange format and the BeCalm framework (http://www.becalm.eu/). The track is designed to foster the development of an integrated and interoperable workflow of multiple text mining tools for real-world testing in pilot publishing frameworks.

The track consists of two phases: 1) a batch phase, where the task is to annotate text from figure legends with the entity types and IDs for taxon (organism), gene, protein, miRNA, small molecules, cellular components, cell types and cell lines, tissues and organs; and 2) an interactive annotation phase, where curators can make use of the system-supplied annotations, to determine whether this speeds the curation

This work was supported under NIH grant 5R01GM080646-11S1 and NIH R13 GM109648.

process. This report describes the batch testing results. Experiments on the interactive annotation phase are now underway.

III. DATA

Training materials were drawn from annotated figure panel legends from SourceData (3). SourceData is working with publishers and authors to create machine-readable descriptions of underlying data in figures and figure legends. By referring to established public databases of biological terms, the specific biological entities, their roles as target, intervention or outcome measure in each paper can be consistently identified. SourceData (TL, RL) made available a large set of curated figure panel legends for the BioID track; the SourceData team made these data sets available "as is", to provide a "public" data set for training and a "private" (not previously released) reference data set for testing system performance.

The track organizers (DC, SB) converted these annotated panel captions into BioC format, which were provided to participants, along with the corresponding full text articles (also BioC format) for context. Participants participated by submitting annotations for one or more of the bioentity types, in BioC format for the set of captions.

A. Bio-ID Training and Test Data Sets

The training data set consisted of 13,573 annotated figure panel captions corresponding to 3,658 figures from 570 full length articles from some 22 journals, for a total of 102,717 annotations. Table 1 shows the distribution of entity types across the training corpus. The test data set consisted of 4,310 annotated figure panel captions from 1,154 figures taken from 196 full length journal articles, with 30,286 annotations in total. This corresponds to roughly 6+ figures/document and 3-4 panels per figure with some 7 annotations per panel caption.

TABLE 1 DESCRIPTION OF TRAINING DATA BY ENTITY TYPE

Entity Type	# articles w entity	Total # by entity type	# unique by entity type
Protein/Gene (561 articles)	UniProt (548)	30211	2833
	NCBI gene (537)	21766	2451
miRNA	9	167	13
Small molecules (513 articles)	ChEBI (506)	9869	786
	PubChem (134)	700	140
Cellular component	482	7310	376
Cell types and cell lines (482 articles)	Cellosaurus (351)	5783	230
	Cell Ontology (300)	4638	217
Tissues & organs	316	5870	459
Organisms & species	454	7952	147

B. Data Set Content

The training data contains a set of files in BioC format. Each file consists of a collection of figure captions for a given article annotated by SourceData curators. The name of the file corresponds to its PubMed Central ID and the corresponding full-text article is provided separately. Within a BioC file, each <document> element contains the annotation for each individual figure panel within the article; the <id> given is made up of the PMCID followed by the Figure_number-panel (e.g., <id>5048346 Figure_1-A</id>), as shown in the example in Fig. 1.

Note that the text for the panel may consist of discontinuous text derived from the Figure caption; it may also include legends from several panels.

C. Annotation Guidelines

Below are important considerations of the curation practice (extracted from the SourceData definitions for curators by Thomas Lemberger):

- The SourceData description of the data presented in scientific figures specifies the entities that are relevant to the scientific meaning of the data. To be tagged by curators, a figure panel must report experimental data. Panels that present schematics, computational simulation results, overviews and workflows are not tagged.
- In the text of a relevant panel legend, all mentions of terms that correspond to entities types on Table 2 are tagged.
- Entities are assigned to only one entity type of those listed in Table 2.
- In general, generic terms referring to broad classes of biological components (e.g. 'proteins', 'cells', 'animals') are not be tagged unless they refer to the object of an assay.

```
[<document>
  <id>5048346 Figure_1-A</id>
  <infon key="sourcedata_document">2225</infon>
  <infon key="doi">10.15252/embj.201694885</infon>
  <infon key="pmc_id">5048346</infon>
  <infon key="figure">Figure 1-A</infon>
  <infon
key="sourcedata_figure_dir">Figure_1-A</infon>
  <passage>
    <offset>0</offset>
    <text>A. The localization of NSUN3 was analysed in
HEK293 cells stably expressing NSUN3-GFP (green).
NSUN3-GFP and staining with a Mitotracker (red) are
shown separately and in an overlay with DAPI to indicate
nuclei. The scale bar represents 5 m.</text>
    <annotation id="1">
      <infon key="type">Uniprot:Q9H649</infon>
      <infon key="sourcedata_figure_annot_id">1</infon>
      <infon
key="sourcedata_article_annot_id">1</infon>
      <location offset="23" length="5"/>
      <text>NSUN3</text>
    </annotation>
    …
```

Fig. 1. Example of BioC training data file

TABLE 2 ENTITY TYPES WITH CORRESPONDING RESOURCES FPR LINKAGE (NORMALIZATION)

Entity Type	Resources	Example infon key="type"	Generic infon key="type"
Protein	UniProt	Uniprot:Q9H649	Uniprot:<Accession>
Gene	NCBI gene	NCBI gene:4137	NCBI gene:<ID>
miRNA	Rfam	Rfam:RF00076	Rfam:<ID>
Small molecules	ChEBI (primary)	CHEBI:15996	CHEBI:<ID>
	PubChem (secondary)	PubChem:5717066	PubChem:<id>
Cellular component	GO cellular component	GO:0005886	<GOID>
Cell types and cell lines	Cellosaurus (primary)	CVCL_U985	<accession>
	Cell Ontology (secondary)	CL:0000540	<CLID>
Tissues & organs	Uberon	Uberon:UBERON:0002048	Uberon:<UBERONID>
Organisms & species	NCBI Taxonomy	NCBI taxon:10090	NCBI taxon:<ID>

Sources: All ID sources are publicly available:
UniProt: www.uniprot.org
Rfam: http://rfam.xfam.org/
PubChem: https://pubchem.ncbi.nlm.nih.gov/
Cellosaurus: http://web.expasy.org/cellosaurus/
Uberon: http://uberon.github.io/
NCBI gene: https://www.ncbi.nlm.nih.gov/gene
ChEBI: https://www.ebi.ac.uk/chebi/
Gene ontology: http://geneontology.org/
Cell Ontology: http://obofoundry.org/ontology/cl.html
NCBI Taxonomy: https://www.ncbi.nlm.nih.gov/taxonomy

• Some terms, such as those referring to proteins or genes, can be appended with prefixes or suffixes that indicate a post-translational modification, a mutation or other variations of the actual base term. In such cases, prefixes or suffixes are left out and only the base term is tagged (e.g., in text describing a mutant of B-RAF 'B-RAF(V600E)', only B-RAF is tagged; similarly with p-AKT1 that designates the phosphorylated form of AKT1, only AKT1 is tagged).

• In other cases, a prefix is added to an entity to denote a species origin, in which case the prefix should be kept (e.g., dMyc to denote the homolog in Drosophila of Myc)

• Some components are engineered by assembling or fusing multiple sub-components; these are tagged individually. For example, the term 'RAS-GFP' referring to a fusion protein between GFP and RAS is annotated with two tags: 'RAS' and 'GFP'.

D. *Special considerations for the BIO-ID track*

• Participating teams can provide annotation for one or more types of those described in Table 1.

• For this track, the gene/protein type can be treated interchangeably (i.e., all proteins and gene mentions can be linked to NCBI gene and/or UniProt identifiers).

• Some entity types, like small molecules, have more than one resource listed; however, one of the resources is considered primary. To help with training and comparison, we provide mappings, to the extent possible, between identifiers of the same entity type. However, the best practice would be to look up the entity in the preferred resource, and if it cannot be found, then look it up in the secondary resource.

• In particular, for UniProtKB, UniProtKB/Swiss-Prot (reviewed) entries should be linked whenever possible; TrEMBL (unreviewed) entries should only be linked when no corresponding Swiss-Prot entry is available.

IV. SCORING

The scorer reports scores at three levels: the individual caption level, the document level, and the corpus level. At the last two levels, scores are aggregated per bioentity type from the previous level. Scoring is done both at the mention level (every occurrence of a bioentity is tagged for its type) and at the normalized identifier level, where the set of unique identifiers in a caption are compared to the reference set of identifiers from SourceData.

Mention level annotations are divided into two classes: the first class consists of annotations which are linked to a biological resource (the "normalized annotations", e.g., in Fig. 1, "NSUN3" is annotated with "Uniprot:Q9H649"). The second class consists of annotations that are associated with an entity type, but not linked to a biological resource, for example, in the text "GFP is green and Calbindin staining is red.", "GFP" is tagged as "protein:GFP" which labels GFP as a protein, but does not link it to an identifier in a specific resource.

The scorer reports mention level scores in 4 conditions:

Score Types	All Entities	Normalizable Only
Exact Match	Any Exact	Norm Exact
Overlapping Match	Any Overlap	Norm Overlap

Exact span match requires that the span (in byte offset) match exactly against the SourceData reference standard; span overlap relaxes this condition to allow a match if the span overlaps with the reference annotation at all. This means that the "overlap" score will be greater than or equal to the "exact" score. The scorer computes mention-level recall/precision/f1-measure for each of these four conditions.

The scorer also computes recall/precision/F1-measure on the normalized IDs which are found, both micro-averaged over the corpus and macro-averaged by document. For the

TABLE 3 TOP SCORING RUNS BY ENITYT TYPE FOR MENTION F1 SCORE ("OVERLAP ALL") AND MICRO-AVERAGED NORMALIZATION F1

	Mention - All Overlap				Normalization Micro Avg				#	#
Entity Type	Team	P	R	F1	Team	P	R	F1	Teams	Runs
cell_type_or_line	407	0.84	0.76	0.80	422_2	0.78	0.56	0.65	2	3
cellular_component	407	0.73	0.55	0.63	422_2	0.55	0.45	0.49	2	3
gene_or_protein	407	0.83	0.84	0.83	407	0.47	0.34	0.40	2	3
organism_or_species	407	0.88	0.83	0.85	393	0.66	0.88	0.76	5	9
small_molecule	407	0.80	0.60	0.69	422_2	0.59	0.47	0.52	3	5
tissue_or_organ	407	0.79	0.63	0.70	407	0.53	0.49	0.51	2	3
miRNA	386	not scored					not scored		1	1

comparison of results, we report here on exact and overlap matching against all annotations for mention level scoring; and micro-averaged scoring for the normalized identifiers.

V. RESULTS

A total of six teams participated. Two teams provided annotations for all six entity types; two teams provided annotations for species only; one team did small molecules only, and one team did miRNA (not scored) and species. The results for the top scoring runs are shown in Table 3, along with the number of teams and runs for each bioentity type. The top scores were selected based on the top F1 score for the "Overlap All" computation for the mention level score. The micro-averaged F1 was used to rank the normalization scores. The set of scores (all types of mention level scores plus micro-averaged normalization scores) for all runs is shown in the Appendix; the high score in each category is highlighted with a green background.

The highest performance was achieved for organism/species, which also had the most participating teams (see Table 3 and Figs. 2 and 3). Fig. 4 shows the results (at both mention level and normalization) for gene/protein as well as organism. Determining the species is a necessary step to determine a correct gene or protein identifier, since these are dependent on species. However, the species may not be explicitly mentioned in the figure caption, making it necessary to use the full text to determine the species or organism for each experiment. We see that the species scores are quite high (both for mention level and normalized micro-average – grey and gold stars). For gene/protein mention, the "Any Overlap" scores (black triangles) are also quite good, but the "Norm Overlap" scores (green triangles) are less good, and the micro-averaged gene/protein normalization scores (red triangles) are much worse – indeed Fig. 3 shows that results for gene/protein normalization are the lowest among the bioentities. One issue is that correct normalization requires stripping of complex prefixes and suffixes from gene names – a challenging task. A second issue is that only two teams tackled the gene/protein task – and these were not the high-scoring systems for species. The results suggest that progress is being made at the mention level, but that gene/protein normalization remains challenging.

ACKNOWLEDGMENTS

The BioID Track has been supported by NIH grants 5R01GM080646-11S1 and NIH R13 GM109648. LH and SB have been supported under MITRE's Internal Research and Development Program.

REFERENCES

1. Lee, C., and Tibbo, H. (2007) Digital Curation and Trusted Repositories: Steps Toward Success. Journal of Digital Information 8, 2

2. https://datascience.nih.gov/

3. Liechti, R., George, N., El-Gebali, S., Götz,,L., Crespo, I., Xenarios, I., and Lemberger, T. (2016) SourceData - a semantic platform for curating and searching figures. preprint in BioRxiv doi: https://doi.org/10.1101/058529

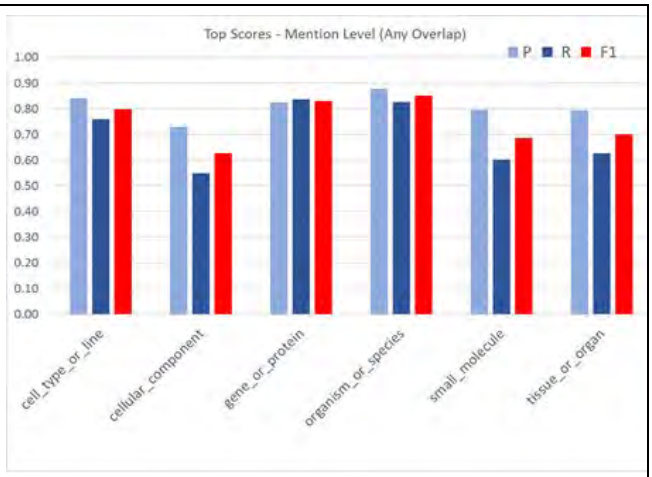


Figure 2: Best Mention Level Scores (Any Overlap) for Entities

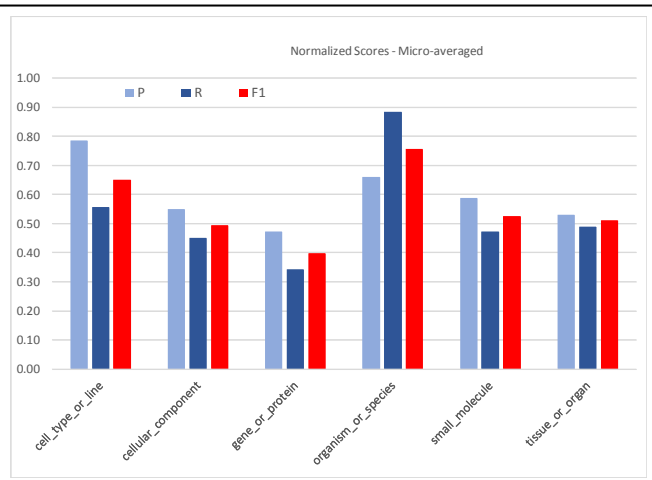


Figure 3: Best Normalized Scores (Micro-Averaged) for Entities

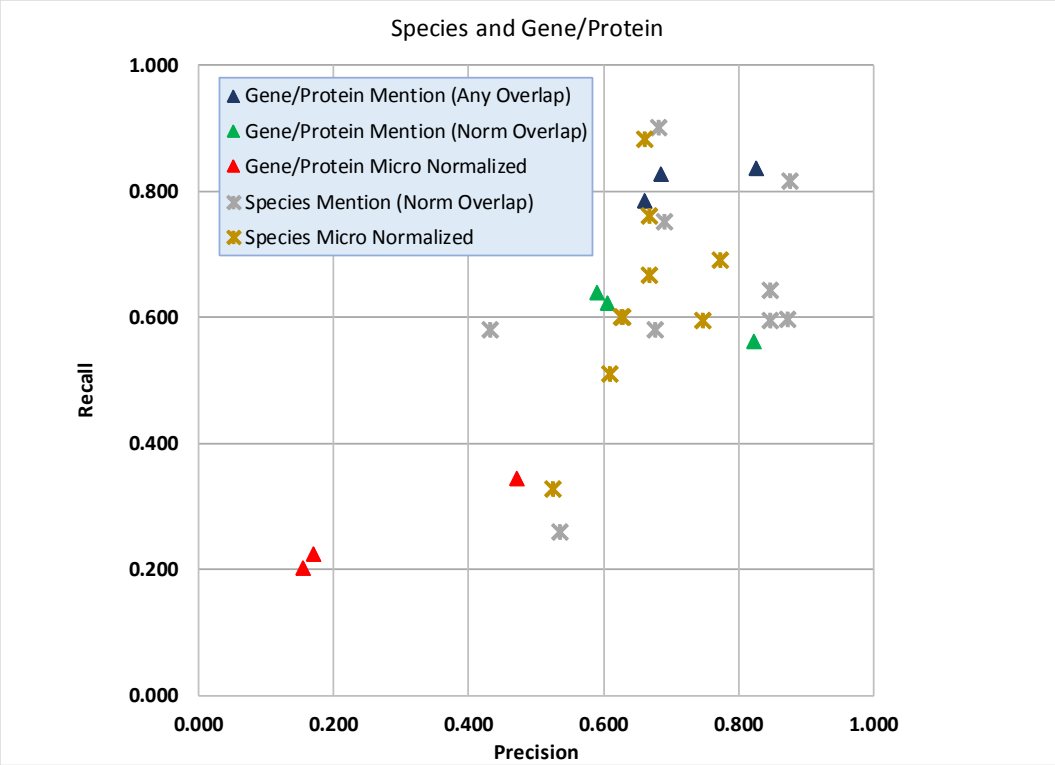


Figure 4: Gene/Protein and Species Results

Appendix: Scores for Mention Level and Micro-Averaged Normalized Scores for all Runs

Entity Type	Run	Any Exact P	Any Exact R	Any Exact F1	Any Overlap P	Any Overlap R	Any Overlap F1	Norm Exact P	Norm Exact R	Norm Exact F1	Norm Overlap P	Norm Overlap R	Norm Overlap F1	Micro Norm P	Micro Norm R	Micro Norm F1
cell type or line	407	**0.783**	**0.708**	**0.744**	**0.841**	0.761	**0.799**	0.774	**0.699**	**0.735**	0.830	**0.750**	**0.788**	0.600	**0.576**	0.588
	422_1	0.563	0.493	0.526	0.764	0.670	0.714	0.638	0.430	0.513	0.827	0.557	0.666	0.686	0.505	0.582
	422_2	0.653	0.650	0.652	0.766	**0.763**	0.765	**0.822**	0.551	0.660	**0.866**	0.581	0.695	**0.784**	0.557	**0.651**
cellular compo-nent	407	**0.673**	**0.508**	**0.579**	**0.728**	**0.550**	**0.627**	**0.685**	**0.482**	**0.566**	**0.737**	**0.519**	**0.609**	0.456	0.371	0.410
	422_1	0.400	0.353	0.375	0.600	0.528	0.562	0.424	0.345	0.380	0.629	0.511	0.563	0.472	**0.467**	0.469
	422_2	0.548	0.439	0.488	0.629	0.505	0.560	0.456	0.319	0.376	0.659	0.461	0.543	**0.550**	0.450	**0.495**
gene or protein	407	**0.729**	**0.739**	**0.734**	**0.825**	**0.836**	**0.831**	0.795	0.543	**0.645**	**0.823**	0.561	**0.667**	**0.472**	**0.343**	**0.397**
	422_1	0.412	0.490	0.447	0.660	0.785	0.717	0.437	0.474	0.455	0.590	**0.640**	0.614	0.154	0.202	0.175
	422_2	0.509	0.613	0.556	0.686	0.826	0.749	0.456	0.469	0.463	0.605	0.622	0.614	0.170	0.224	0.193
organism or species	386_1	0.668	0.572	0.616	0.677	0.580	0.625	0.667	0.572	0.616	0.677	0.580	0.625	0.628	0.601	0.614
	386_2	0.426	0.572	0.489	0.433	0.580	0.496	0.426	0.572	0.488	0.432	0.580	0.495	0.626	0.601	0.613
	393_1	0.663	0.874	0.754	0.683	**0.900**	0.776	0.662	0.874	0.754	0.682	0.900	0.776	0.660	**0.883**	**0.756**
	393_2	0.671	0.731	0.699	0.690	0.752	0.720	0.670	0.731	0.699	0.689	0.752	0.719	0.668	0.760	0.711
	393_3	0.516	0.251	0.337	0.536	0.260	0.350	0.516	0.251	0.338	0.536	0.260	0.350	0.526	0.327	0.403
	407	**0.860**	**0.809**	**0.834**	**0.878**	0.826	**0.852**	**0.857**	**0.797**	**0.826**	**0.877**	**0.815**	**0.845**	0.668	0.667	0.667
	408	0.813	0.616	0.701	0.848	0.642	0.731	0.812	0.616	0.701	0.847	0.642	0.730	0.609	0.510	0.555
	422_1	0.649	0.496	0.563	0.837	0.640	0.726	0.686	0.483	0.567	0.846	0.595	0.699	0.747	0.594	0.662
	422_2	0.746	0.715	0.730	0.814	0.780	0.796	0.723	0.495	0.588	0.872	0.597	0.709	0.772	0.691	0.729
small molecule	407	**0.775**	**0.587**	**0.668**	0.796	**0.603**	**0.686**	**0.787**	**0.581**	**0.668**	0.803	**0.593**	**0.682**	0.244	0.240	0.242
	422_1	0.503	0.359	0.419	0.697	0.496	0.579	0.504	0.326	0.396	0.696	0.451	0.547	0.587	**0.473**	**0.524**
	422_2	0.562	0.451	0.500	0.683	0.548	0.608	0.490	0.331	0.395	0.686	0.463	0.553	0.654	0.394	0.492
	425_1	0.584	0.065	0.116	**0.915**	0.101	0.182	0.582	0.069	0.123	**0.910**	0.107	0.192	**0.901**	0.150	0.257
	425_2	0.563	0.050	0.091	0.902	0.079	0.146	0.538	0.051	0.092	0.856	0.080	0.147	0.791	0.102	0.180
tissue or organ	407	**0.727**	**0.575**	**0.643**	**0.793**	**0.627**	**0.701**	**0.604**	**0.542**	**0.572**	0.669	**0.600**	**0.632**	0.531	**0.490**	**0.510**
	422_1	0.547	0.405	0.465	0.741	0.548	0.630	0.519	0.394	0.448	**0.678**	0.515	0.586	0.547	0.434	0.484
	422_2	0.572	0.559	0.565	0.671	0.654	0.662	0.497	0.417	0.453	0.654	0.548	0.596	**0.584**	0.442	0.503

A Study on Identification of Organism and micro-RNA Mentions in Figure Captions

Nai-Wen Chang[1,2], Jitendra Jonnagaddala [3,4], Feng-Duo Wang[5], Hong-Jie Dai[5,6*]
[1]Graduate Institute of Biomedical Electronics and Bioinformatics, National Taiwan University, Taipei, Taiwan
[2]Institute of Information Science, Academia Sinica, Taipei, Taiwan
[3]School of Public Health and Community Medicine, UNSW Sydney, Australia
[4]Prince of Wales Clinical School, UNSW Sydney, Australia
[5]Department of Computer Science and Information Engineering, National Taitung University, Taitung, Taiwan
[6]Interdisciplinary Program of Green and Information Technology, National Taitung University, Taitung, Taiwan

***Abstract*—**The detection of organism mentions in scientific literature facilitates researchers with the ability to find relevant subsets of papers based on species-specific queries. Furthermore, most biological articles will describe pathways or regulation information in figure captions to enhance the understanding of experimental results. The extraction of miRNA and organism from figure captions is useful in characterizing the research studies. In this study, we adopted openly available organism recognition tools and our statistical principle-based miRNA recognizer for identifying organism and miRNA mentions in figure captions of an article. The miRNA recognizer is extended by generating scores for matched slots and indexes for matched terms to normalize recognized miRNAs to identifiers in the Rfam database. We study the performance of the existing tools in recognizing terms in figure captions and the challenges remained to address by evaluating them on the BioCreative VI Bio-ID dataset. We believe the Bio-ID corpus provide a nice starting point for evaluating the performance of miRNA normalization system. In the future, we would like to undertake more comprehensive evaluation of existing tools for extraction of organism/species and would like to enhance the consistency and comprehensiveness of miRNA annotations in the dataset.

Keywords*—*biomedical text mining; organism; micro-RNA; statistical principle-based approach

I. INTRODUCTION

The term organism is used to represent an important class of entities frequently mentioned in biomedical literature. Normalizing organism terms mentioned in literature to database records, such as NCBI taxonomy can be used for disambiguating biomedical entities such as mutations, proteins or genes [1]. Primarily, organism names are based on a well established hierarchical nomenclature conventions developed from the 18th century. However, the recognition of taxonomic groups in biomedical texts such as PubMed articles present a number of issues and challenges. Especially, there is a great degree of ambiguity in the way taxonomic information is expressed in biomedical literature. Large numbers of abbreviations of species names and use of common English names instead of Latin names are the primary reasons for this ambiguity. The use of acronyms, which can be both species specific and species independent, also pose a challenging problem for information extraction tasks. Lastly, incorrect spellings are often an issue with biomedical texts.

MicroRNAs (miRNAs) have become one of the hottest subjects in science and medicine recently. The first formal recognition of miRNAs was ten years ago. Since then miRNAs have been found to have a critical role in regulating many physiological processes and pathological processes. Numerous miRNAs and their associated targets have been identified by bioinformatics tools [2-4] and high-throughput sequencing [5-7]. Therefore, the demand of recognizing miRNAs mentioned in literature is increasing. miRNAs are evolutionary-conserved small non-coding RNA molecules that post-transcriptionally regulate gene expression by base-pairing to messenger RNAs (mRNAs). Many freely available, web-based miRNA-related database systems have been developed for researchers to retrieve miRNAs and their target genes. For instance, miR2Disease is a manually curated database, providing a comprehensive resource for miRNA deregulation in various human diseases [8]. miR2Disease provides researchers with information such as miRNA-disease relationships and experimentally verified miRNA-target genes, as well as references to the relevant biomedical literature. Similarly, the miRWalk database provides predicted and validated miRNA binding site information related to miRNAs in humans, mice and rats [9].

Information extraction methods can be employed to extract organism and miRNA related information. The identification of these two entities can facilitate taxonomy-aware information extraction in construction of valuable knowledge bases such as miRWalk and miR2Disease. In addition, these methods can also be used to enhance the index created by search tools for retrieving more relevant literature using species-specific and miRNA-related keywords. Furthermore, most biological articles will describe pathways or regulation information in figure captions to enhance the experimental results. The advantages of extracting miRNA and organism from figure captions can capture the most important and real data. With this in mind, we assessed the current automated information extraction methods available to extract organism information from figure captions

* Corresponding author

We also extended our previous miRNA recognition method [10] by developing a method for normalizing recognized terms to the Rfam database.

II. Methods

A. Dataset

We used the dataset released by Bio-ID track for our research purpose. The dataset was prepared as part of the EMBO SourceData project[2], which consisted documents in BioC [11] format with number of figure captions from full-length articles along with the annotations for multiple bio-entities. Organisms and miRNAs are two of the entity types annotated in this dataset and the annotations includes their spans in figure captions and their corresponding database IDs. This dataset was used to assess the performance of several current openly available tools. We employed these tools for extract organism/species information from figure captions to understand the complexity, issues and challenges involved in that.

B. Species Entity Recognition and Normalization

For recognition and normalization of organism/species entities we consider SR4GN [1], ORGANISM/SPECIES tool [12], and NCBO Annotator [13]. After studying their performance on the training set, we decided to use the SPECIES tool for processing the test set of the Bio-ID track.

The SPECIES tool identifies and normalizes species entities using dictionary look-up approach. This tool employs NCBI Taxonomy for dictionary look-up. A minor post-processing enhancement was developed in this work. The enhancement primarily involved selection of top 10 entities with highest frequency observed in the training set. Once these entities are identified, they are checked against the output (on the test set) from SPECIES tool where an entity is observed with no NCBI taxonomy ID assigned. Should there be any such entity, NCBI taxonomy ID is assigned based on the top 10 entities list prepared earlier. This process was mainly employed to improve the performance but also at the same time limit the number of false positives that may creep in because of this post processing enhancement.

C. miRNA entity Recognition and Normalization

Rule-based and machine learning-based approaches are two popular methods used in the task of miRNA recognition, which have been applied to public miRNA databases, such as miRCancer, miRSel [13, 14] and TarBase [15, 16]. However, rule-based approaches require explicit rules developed by domain experts, which are not flexible enough to cover all variations, such as the insertion, deletion or substitution (IDS) of words appearing in the entities, phrases or sentences. On the other hand, machine learning models can learn the implicit patterns automatically, but the model cannot be easily interpreted by humans. We have proposed the statistical principle-based approach (SPBA) for miRNA recognition to deal with the drawbacks of the above approaches [10]. In general, SPBA can automatically extract labeled sequences, combine them into more representative principles through the observation of dominated principles, and employ a partial matching algorithm to harness the advantages of both rule- and machine learning-based approaches while surpassing their limitations. The performance of the SPBA-based miRNA recognition is evaluated on the corpus annotated by Bagewadi, Bobić, Hofmann-Apitius, Fluck and Klinger [14]. The evaluation achieved a 0.988 F-score, 0.986 precision (P) and 0.991 recall (R), which outperformed the traditional rule-based methods. However, this method only considers the recognition task. In this study, we extended the existing method to support the normalization process that can link recognized miRNA mentions to identifiers in the Rfam database, a database of non-coding RNA families and other structured RNA elements. The details of SPBA is described in the following subsection.

D. Knowledge Construction for miRNA Recognition

Our SPBA-based method used a collection of principles generated from the training phase to match the content. If the content can be matched with a compiled labeled sequence, the corresponding entity is determined. The training phase of SPBA consists of three main steps. The first is knowledge construction. In case of miRNA recognition, we represent the knowledge related to miRNA through semantic slots and principles semi-automatically. Following is the principle generation step, in which slots are assembled and summarized by observing the arrangement of principle slots which can accomplish the target task. Lastly, a flexible principle matching algorithm allowing insertion/deletion/substitution is applied to extract the information represented by the compiled principles in unstructured text. Fig. 1 illustrates a simplified example of how the knowledge was constructed for representing a miRNA in SPBA under the principle-slot scheme. More precisely, the knowledge is constructed in a hierarchical structure.

In the knowledge representation of SPBA, the root node is usually the name of a domain or a subject. In Fig 1. the root node is "miRNA" indicating that the structure represents the knowledge for miRNA names. The first child node of a root node is usually the "SLOT" node, under which we can define the fundamental slot for miRNA. Albeit the heterogeneous writing styles, some common contents can be found among miRNAs, which form the backbone of miRNA's slots. For instance, both the miRNA "cel-miR-123-5p" and "hsa-microRNA-24-3P" consists of a species (cel and hsa), the indicating word "miRNA" and a hairpin that possess unique feature in representing a miRNA. Hence, they can be designated using the following combination of slots "[Species][miRNA][order][Hair-pin]". Here we use brackets to enclose a slot name for representing a slot. For example, "[Species]" is a slot that encodes the species in which the miRNA appears. "[miRNA]" is the slot representing the word indicating an occurrence of a miRNA name.

The last two slots can be further generalized into one slot, "[Suffix]", which can be used to differentiate distinctive types between miRNAs (e.g., has-let-7a-2-3p). Therefore, they are organized in a hierarchical structure as showed in Fig 1 (3: [Suffix] → SLOT → [Hair-pin]). For each slot, a list of terms that could be written in literature are collected and listed under that slot. For instance, the instances of the "[Species]" slot are

[2] http://sourcedata.embo.org/

tri-grams, such as "hsa" and "cel". The indicating words for [miRNA] include "mir", "let", "lsy", "micro RNA", etc.

E. Principle Matching

During the principle alignment procedure, we score those possible candidate principles based on matched slots, slot relations and insertions. Each exactly matched slot gets a score of 4. If there are insertion/deletion/substitution in the string, the scoring mechanism will assign scores accordingly. We calculate the score of an insertion by gathering its left (resp. right) bigram statistics with its neighboring left (resp. right) slots in the training set. The bigram frequency gives a way to assign the insertion scores, which usually fall in the ranges, -(∞), -2, -1, 0, 1, and 2. Deletions are assigned a score of -(∞), -2, -1, or 0. A substitution is either a partial match or a category match of the slot, which is usually assigned a score of 1 or 2. The final score of a principle is the sum of all the scores of this principle. The length of a principle, which means the number of slots of a principle, is used as the threshold to determine whether this principle is matched or not. Finally, the longest principle or a principle which contains the most slots will be considered as matched. In other words, the principles will be ignored if the final score closes to -(∞).

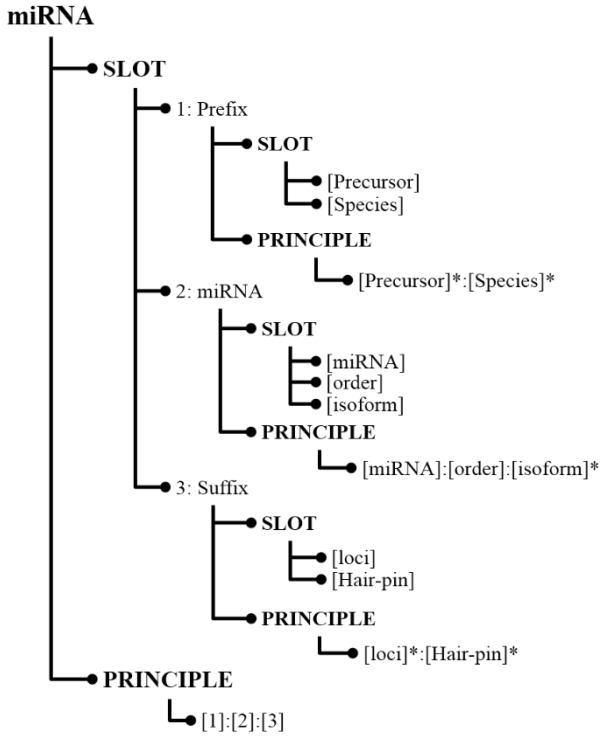


Fig. 1. Knowledge represented for miRNA in SPBA.

F. Principle-based Normalization

For normalization, we first downloaded the family file from ftp://ftp.ebi.ac.uk/pub/databases/Rfam/CURRENT/database_files. We extracted the following columns to compile the lexicon for normalization:

- The first column: the family accession number (e.g. RF00994).
- The second column: the family id (e.g. mir-1255)
- The fourth column: the family description (e.g. microRNA mir-1255)

We then used the generated principles to match all columns contained miRNA names. During the principle matching process, we scored the matched slots based on the matched principles over all entries in the compiled lexicon and built indexes for each slot. Therefore, each slot will associate with all possible corresponding grounding entries in our lexicon along with a matching score.

After the principle matching step, we were able to recognize possible miRNA mentions. For a recognized miRNA mention, the indexes of its matched slots were used to retrieve all possible grounding in the Rfam database. We then used the associated score to select the normalization ID.

III. Results

A. Entiy Recognition Performance on the miRNA Interaction Corpus

We first report the performance of the developed miRNA recognizer on the corpus annotated by Bagewadi, Bobic, Hofmann-Apitius, Fluck and Klinger [15]. This corpus only contains annotations for the spans of miRNAs that appeared in literature. Therefore, we can only estimate the entity recognition performance. They distinguished their annotations for miRNA into two types: specific-miRNA (*e.g.* has-miR-124b), and non-specific-miRNA (*e.g.* microRNAs or miRNAs). In our evaluation, we only focused on the specific-miRNA type.

TABLE I. Entiy Recognition Performance on the miRNA corpus

	Training set	Test set
Precision	0.994	0.986
Recall	0.9902	0.991
F-score	0.992	0.988

As shown in Table I, our tool can achieve a precision (P) of 0.994, recall (R) of 0.9902, and F-score (F) of 0.992 in the training set. Moreover, the performances in the test corpus are a precision of 0.986, recall of 0.991, and F-score of 0.988.

B. Entity Normalization Performance on the Bio-ID corpus

Table II shows the performance of normalization on the training set of the Bio-ID track. We report the performance in terms of micro-Precision, Recall and F-measure under the overlapping mode. We observed that the extended methods achieved recall of 0.865 on the training set but a very lower precision of 0.253 resulting in a frustrated F-score of 0.373 on the Bio-ID dataset. Although miRNA was annotated in the Bio-ID corpus, the test set was not. Thus, we cannot report our performance on the test set. After analyzing the errors on the training set, we observed that the majority of the errors are due to inconsistent annotations. For instance, U2 (Rfam:RF00004) mentioned several times in the Figure 4 of the article (PMC4801943) was not annotated in the corpus. However, our method recognized and normalized that entity.

Table II also shows the performance of the three off-the-shelf organism/species identification tools on the training and test sets. The performance was evaluated under overlap setting only for normalization component of the tools. We can observe that NCBO annotator had the best R but a very low P. The SPECIES tool achieves the best F-score. Therefore we selected SPECIES tool as a baseline system to study its performance on the test set. The SPECIES tool was employed under two different settings (Run 1 and Run 2). Under the Run 1 setting, the tool was executed on the Bio-ID corpus with default configuration. In Run 2, the post-processing enhancement described in the Methods section was applied. The performance of SPECIES tool in Run 1, with default configuration is better than Run 2, where the post-processing enhancement didn't impact R but decreased the overall F-score.

TABLE II. NORMALIZATION PERFORMANCE

Configuration	Train set			Test set		
	P	**R**	**F**	**P**	**R**	**F**
miRNA	0.253	0.865	0.373	n/a	n/a	n/a
NCBO Annotator	0.061	**0.920**	0.118	n/a	n/a	n/a
SR4GN	0.468	0.382	0.419	n/a	n/a	n/a
SPECIES-Run1	**0.670**	0.476	**0.557**	**0.677**	**0.580**	**0.625**
SPECIES-Run2	0.460	**0.481**	0.471	0.432	**0.580**	0.495

IV. CONCLUSION

In this study, we presented performance assessment of miRNA and organism information extraction tools with focus on normalization aspect, using two different datasets. We believe the results presented in this study provide a good starting point for evaluating the performance of miRNA and organism entity recognition and normalization system. In future, we would like to improve the manual annotations in these two datasets. Specifically, we would like to enhance the consistency and comprehensiveness of the miRNA annotations. Additionally, we would also like to undertake more comprehensive evaluation of existing tools for extraction of organism/species and miRNA related information.

ACKNOWLEDGMENT

This study was conducted as part of the electronic Practice Based Research Network (ePBRN) and Translational Cancer research network (TCRN) research programs. ePBRN is funded in part by the School of Public Health & Community Medicine, Ingham Institute for Applied Medical Research, UNSW Medicine and South West Sydney Local Health District. TCRN is funded by Cancer Institute of New South Wales and Prince of Wales Clinical School, UNSW Medicine. This study was also supported by the Ministry of Science and Technology of Taiwan [grant numbers MOST-105-2221-E-143-003 and MOST-106-2221-E-143-007-MY3].

REFERENCES

[1] C.-H. Wei, H.-Y. Kao, and Z. Lu, "SR4GN: A Species Recognition Software Tool for Gene Normalization," *PLoS ONE,* vol. 7, no. 6, pp. e38460, 2012.

[2] D. M. Garcia, D. Baek, C. Shin, G. W. Bell, A. Grimson, and D. P. Bartel, "Weak seed-pairing stability and high target-site abundance decrease the proficiency of lsy-6 and other microRNAs," *Nat Struct Mol Biol,* vol. 18, no. 10, pp. 1139-46, Oct, 2011.

[3] A. J. Enright, B. John, U. Gaul, T. Tuschl, C. Sander, and D. S. Marks, "MicroRNA targets in Drosophila," *Genome Biol,* vol. 5, no. 1, pp. R1, 2003.

[4] A. Krek, D. Grun, M. N. Poy, R. Wolf, L. Rosenberg, E. J. Epstein, P. MacMenamin, I. da Piedade, K. C. Gunsalus, M. Stoffel, and N. Rajewsky, "Combinatorial microRNA target predictions," *Nat Genet,* vol. 37, no. 5, pp. 495-500, May, 2005.

[5] S. Baskerville, and D. P. Bartel, "Microarray profiling of microRNAs reveals frequent coexpression with neighboring miRNAs and host genes," *RNA,* vol. 11, no. 3, pp. 241-7, Mar, 2005.

[6] M. V. Iorio, M. Ferracin, C. G. Liu, A. Veronese, R. Spizzo, S. Sabbioni, E. Magri, M. Pedriali, M. Fabbri, M. Campiglio, S. Menard, J. P. Palazzo, A. Rosenberg, P. Musiani, S. Volinia, I. Nenci, G. A. Calin, P. Querzoli, M. Negrini, and C. M. Croce, "MicroRNA gene expression deregulation in human breast cancer," *Cancer Research,* vol. 65, no. 16, pp. 7065-7070, Aug 15, 2005.

[7] A. Helwak, G. Kudla, T. Dudnakova, and D. Tollervey, "Mapping the Human miRNA Interactome by CLASH Reveals Frequent Noncanonical Binding," *Cell,* vol. 153, no. 3, pp. 654-665, Apr 25, 2013.

[8] Q. Jiang, Y. Wang, Y. Hao, L. Juan, M. Teng, X. Zhang, M. Li, G. Wang, and Y. Liu, "miR2Disease: a manually curated database for microRNA deregulation in human disease," *Nucleic Acids Res,* vol. 37, no. Database issue, pp. D98-104, Jan, 2009.

[9] H. Dweep, C. Sticht, P. Pandey, and N. Gretz, "miRWalk--database: prediction of possible miRNA binding sites by "walking" the genes of three genomes," *J Biomed Inform,* vol. 44, no. 5, pp. 839-47, Oct, 2011.

[10] N.-W. Chang, H.-J. Dai, Y.-L. Hsieh, and W.-L. Hsu, "Statistical Principle-based Approach for Detecting miRNA-target Gene Interaction Articles," in Proceeding of the IEEE 16th International Conference on BioInformatics and BioEngineering (BIBE), Taichung, Taiwan, 2016.

[11] D. C. Comeau, R. Islamaj Dogan, P. Ciccarese, K. B. Cohen, M. Krallinger, F. Leitner, Z. Lu, Y. Peng, F. Rinaldi, M. Torii, A. Valencia, K. Verspoor, T. C. Wiegers, C. H. Wu, and W. J. Wilbur, "BioC: a minimalist approach to interoperability for biomedical text processing," *Database (Oxford),* vol. 2013, pp. bat064, 2013.

[12] E. Pafilis, S. P. Frankild, L. Fanini, S. Faulwetter, C. Pavloudi, A. Vasileiadou, C. Arvanitidis, and L. J. Jensen, "The SPECIES and ORGANISMS Resources for Fast and Accurate Identification of Taxonomic Names in Text," *PLoS ONE,* vol. 8, no. 6, pp. e65390, 2013.

[13] C. Jonquet, N. H. Shah, and M. A. Musen, "The Open Biomedical Annotator," in AMIA Summit on Translational Bioinformatics, San Francisco, CA, USA, 2009, pp. 56-60.

[14] S. Bagewadi, T. Bobić, M. Hofmann-Apitius, J. Fluck, and R. Klinger, "Detecting miRNA Mentions and Relations in Biomedical Literature," *F1000Research,* vol. 3, no. 205, 2014.

[15] S. Bagewadi, T. Bobic, M. Hofmann-Apitius, J. Fluck, and R. Klinger, "Detecting miRNA Mentions and Relations in Biomedical Literature," *F1000Res,* vol. 3, pp. 205, 2014.

A Neural Named Entity Recognition Approach to Biological Entity Identification

Emily Sheng, Scott Miller, José Luis Ambite, Prem Natarajan
Information Sciences Institute/USC, Marina del Rey, USA

***Abstract*—We approach the BioCreative VI Track 1 task of biological entity identification by focusing on named entity recognition (NER) and linking tagged entities to standard database identifiers. For this task, we apply recent neural NER techniques of combining bi-directional long short term memory (BLSTM) network layers with conditional random fields (CRFs) to the biomedical domain. We then use context words, dictionary lookups, and external biological knowledge bases to match tagged biological entities with corresponding identifiers. Our system predicts cell types and cell lines, cellular components, organisms and species, proteins and genes, small molecules, and tissues and organs.**

Keywords—named entity recognition; NER; bi-directional LSTM; conditional random fields; CRF; dictionary lookup

I. Introduction

In Track 1 of the BioCreative VI tasks, we are asked to automatically identify biological entities in biomedical text and link them to their standard database identifiers (e.g., UniProt ID for proteins and genes, ChEBI ID for small molecules, etc). Given biological entities annotated with identifiers in figure captions from 570 full-length articles as training data, we need to recognize and link biological entities in figure captions from 196 unseen articles to identifiers. Precision, recall, and F-measure at the caption, document, and corpus level for different entities are calculated. Additionally, there are also distinct scores for strict versus overlapping entity boundary matches, and for measuring across all annotations (i.e., including entities linked to a generic term such as "protein" and entities linked to a standard database identifier) versus just normalized entities (i.e., only entities linked to an identifier). It is useful to work with figure captions because sentences in captions describe figure objects, which are often the biological entities of interest. Also, extracting textual information from figure captions allows us to potentially link the textual data with figure data. Having methods to automatically extract and ground entities would be beneficial to the progression of research in scientific communities.

We train a CRF-based model [4] using NERSuite [3] as an NER baseline and compare it to the neural model. Although we are exploring different neural NER architectures, time constraints dictate that we use the LSTM-CRF architecture described in [5] as the NER model in our task submission. Our entity grounding component is based on dictionary and API lookups, and we apply some heuristics to more accurately segment entities for grounding.

This work was supported in part by the DARPA Big Mechanism program (W911NF-14-1-0364).

II. Preprocessing

We do as little preprocessing as possible to keep this component of the pipeline simple, generalizable, and easy to re-assemble into the BioC output format. For each caption paragraph, we use the NTLK (http://www.nltk.org/) sentence tokenizer to extract individual sentences, and then split each sentence on whitespace into "words" that we attach annotations to in the CoNLL format. Note that although we refer to the tokens divided by whitespace in a sentence as "words", these "words" can contain varying amounts of punctuation and are not necessarily well-formed English words. If the last byte in a sentence is ".", "?", or "!", we separate it into an additional "word". NERSuite takes a file with one sentence per line as input, while the neural NER model takes data in the CoNLL format as input. Both models output a tag for each "word" in the IOB format [7]. Our simple data tokenization into "words" and "word"-level tags means that there could potentially be multiple ground truth entities and entity types in one model-tagged entity or extraneous characters in model-tagged entities. If there are multiple annotations for a "word", we take the first annotation of an entity the word belongs to as the ground truth, and ignore all subsequent annotations that include this "word". We take sentences from a random 80% of the 570 articles to form the training set, sentences from 10% of the articles to form the development set, and sentences from the remaining 10% of the articles to form the test set.

TABLE I. Distribution of annotated entity types in training, development, and test sets

Entity type	Training set	Dev set	Test set
gene_or_protein	39,190 (54.41%)	4,312 (50.73%)	4,945 (57.71%)
small_molecule	8,602 (11.94%)	1,324 (15.58%)	857 (10.00%)
cellular_component	5,970 (8.29%)	530 (6.24%)	617 (7.20%)
cell_type_or_line	8,116 (11.27%)	966 (11.36%)	937 (10.94%)
tissue_or_organ	4,638 (6.44%)	538 (6.33%)	651 (7.60%)
organism_or_species	5,511 (7.65%)	830 (9.76%)	561 (6.55%)
Total	72,027 (100.00%)	8,500 (100.00%)	8,568 (100.00%)

Table 1 shows that a little more than half of the annotated entities across the training, development, and test sets are *gene_or_protein*.

III. Named Entity Recognition

A. NERSuite

We train a CRF-based baseline using NERSuite, so that we may compare the effects of our neural NER approach with a more standard CRF model. NERSuite is a toolkit that uses features derived from a tokenizer, part-of-speech tagger, lemmatizer, chunker, and optionally, dictionaries, as input into a CRF model. For our baseline model, we use all of the standard features except dictionaries to train NERSuite on our training and development set; we report results on the test set.

B. BLSTM-BLSTM-CRF

1) Related work: In recent years, a popular model for NER has been to derive character embeddings from a BLSTM or CNN model, combine the character embeddings with word embeddings and feed the concatenated result into another BLSTM layer. Some works additionally include a CRF layer that takes the output of the BLSTM layer as input. Chiu et al. [2] feed character embeddings and additional character features into a convolutional neural network (CNN) layer, and then concatenate the extracted character representation with word embeddings and additional word features to feed into a BLSTM layer. The BLSTM output is then forwarded to output layers to predict the best sequence of tags for a sentence. Lample et al. [5] concatenate word embeddings and BLSTM-extracted character embeddings to feed into a BLSTM layer, and then feed the BLSTM output to a CRF layer. Ma and Hovy [6] input character embeddings into a CNN layer, and then concatenate the extracted character representation with word embeddings to forward to a BLSTM and then CRF layer.

In our submitted model, we use the architecture shown in Fig. 1 and described in [5].

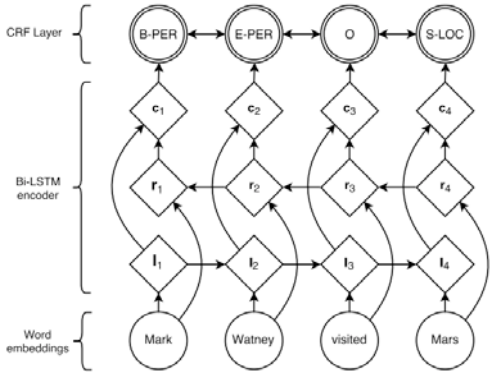


Fig. 1. This figure is taken from [5]. We apply their network with minor changes to the BioCreative dataset.

2) Word embeddings: Many previous works report that using pretrained word embeddings instead of randomly initializing word embeddings can significantly help increase NER scores. In our model, we use word embeddings pretrained on a combination of all abstracts from PubMed, all full-text from PubMed Central (a collection of open access documents from PubMed), and a Wikipedia dump [1]. We observe that these pretrained word embeddings boost the scores across all biological entity types significantly.

3) Character-based representation: To derive a character-based representation of words, we randomly initialize a 25-dimension vector for each character and input it to a BLSTM layer with a 25-dimension hidden layer. We assume, similarly to [5], that the hidden layer values of the last character in a word encode the character-level context of the entire word in a forward LSTM pass. Thus, we concatenate the hidden values of the last character in a word in the forward LSTM pass with the hidden values of the first character in a word in the backward LSTM pass to create the character-based representation of the word.

4) BLSTM-CRF with final word representation: To create an informative word representation, we concatenate the word embedding of a word with the character-based representation of the word. The final word representation is then input into a dropout layer, where half of the values from the word representation are dropped in training. Next, the word representations are input into a BLSTM layer with a hidden dimension of size 200. Similarly to the character-based BLSTM, we concatenate the hidden values of the last word in a sentence in the forward LSTM pass with the hidden values of the first word in a sentence in the backward LSTM pass to form the layer output. We then pass the output to a hidden layer to shrink the vector dimension back down to size 200, and use another hidden layer to shrink the vector to a dimension equal to the number of unique NER tags in the training data using the IOBES annotation scheme. In our training data, there are 25 such unique NER tags. The NER model internally uses the IOBES scheme, which also keeps track of singleton annotations and the end tokens of annotations; the final model output uses the IOB scheme. Lastly, we pass the 25-dimensional output vector to a CRF layer, where the CRF will use the BLSTM output vector and transition scores between a pair of tags to maximize the probability of the groundtruth tag sequence in training.

5) Parameters: The pretrained word embeddings are 200-dimensional vectors, so we use a 200-dimensional hidden layer in the word BLSTM. Lample et al. [5] use 100-dimensional vectors trained on news corpora and a 100-dimensional hidden layer in the word BLSTM. The first hidden layer uses a tanh activation function, and the second hidden layer uses a sigmoid activation function. We use stochastic gradient descent with a learning rate of 0.01 as the optimization function.

IV. Postprocessing

Before passing on tagged entities to be linked to identifiers, we strip all punctuation in the string "!\"#$%'()*+,-./:;<=>?@[\\]^_`{|}~" recursively from the beginning of the tagged entity. We also strip all punctuation in the string

[1] The details of the pretrained word embeddings are at http:/bio.nlplab.org/.

"!\"#$()*+,-./:;<=>?@[\\]^_`{|}~" from the end of the tagged entity. This does not completely remove all extraneous characters from tagged entities, nor does it remove only extraneous characters, but it works as a simple heuristic[2]. We attempt to find multiple proteins or genes that exist within a larger entity tagged as a protein or gene, but otherwise we do not further address the problems of multiple entities or entity types in a tagged entity in this work.

V. Grounding

A. Contextual dictionary

For annotated entities that are part of a larger word in the original sentence (e.g., "Tau" in "EcrTgTaumouse"), the words around the entity ("EcrTg" and "mouse" in the previous example) can be useful context words for linking to a standard identifier. We create a dictionary that maps all annotated entities that occur in the 570 training documents to the list of identifiers that entities have been linked to. Additionally, for each (entity, identifier) pair, we create a list of all the context "words" that are known to be associated. For example, for the entity "Tau", we could have the associated identifiers: "NCBI gene:17762" and "NCBI gene:4137". We further note that when "Tau" is linked to "NCBI gene:17762", the context words "EcrTg" and "mouse" are used. When "Tau" is linked to "NCBI:4137", the context words "EcrTg" and "human" are used. We do not weight context words based on how often they occur for an (entity, identifier) pair; this is left for future work. For each tagged entity that our NER model finds, we first check if the entity is in this contextual dictionary. If so, we link the entity to the known identifier that shares the most contextual words with the sentence the entity belongs to. Taking our example above, if the sentence were "Tau composition in ECrTgTau and control mouse lines investigated," the identifier "NCBI gene:17762" would have more context words in common with the words in the sentence and thus be assigned. Our intuition is that context words are strong indicators of species and other differentiating factors between identifiers of entities with the same surface forms.

B. Searching external knowledge bases

Table 2 shows the knowledge bases, API sources, and generic labels associated with each entity type. If we do not find a tagged entity in the contextual dictionary, we try to search for it in the appropriate external knowledge base. All searches for the submitted model were done on Aug. 22, 2017.

For proteins and genes, we use UniProt's official API[3] to search for entity identifiers. We use NCBI's Entrez tool to search the "taxonomy" database for identifiers for organisms and species [8]. For the rest of the entities, we use AmiGO, which is a collection of tools for searching the Gene Ontology database as well as knowledge bases for a few other ontologies [1]. We assign the first identifier match found in the appropriate knowledge base through the API source. If no matches are returned, and if the entity type is not *gene_or_protein*, we assign the entity to its generic label.

C. Further heuristics for proteins and genes

Proteins and genes have the most number of samples out of all the annotated biological entity types and also numerous variations in surface forms. For example, proteins "Tau", "MAPT", "MAPTL", and "MTBT1" are all synonyms. In contrast, the cellular component "ribosome" has synonyms "ribosomal RNA", "free ribosome", and "membrane bound ribosome", which are more similar to each other in surface form. We manually review some examples of proteins and genes in the training data and devise the following heuristic:

- If the complete tagged entity is not found through searching the UniProt API:
    - If there is whitespace in the tagged entity, split on the whitespace
    - Else if there is a forward slash "/" in the tagged entity, split on the "/"
    - Else if there is a dash "-" in the tagged entity, split on the "-"
    - Else if there is a semicolon ";" in the tagged entity, split on the ";"
- Search each split entity through the UniProt API. If identifiers are found, link the entity to the first found identifier. If no identifiers are found or if there are no whitespace, "/", "-", or ";" characters in the entire entity, just assign the generic label "protein".

TABLE II. Sources of Information Across Entity Types

Entity type	Knowledge base	API source	Generic labels
gene_or_protein	UniProt	UniProt	protein
small_molecule	ChEBI	AmiGO	molecule
cellular_component	GO	AmiGO	subcellular
cell_type_or_line	CL	AmiGO	cell
tissue_or_organ	Uberon	AmiGO	tissue
organism_or_species	NCBI taxon	Entrez (db: taxonomy)	organism

TABLE III. Precision, Recall, and F_1 Scores Across Entities for Different NER Models

Entity type	NERSuite			BLSTM-BLSTM-CRF		
	P	***R***	***F_1***	***P***	***R***	***F_1***
gene_or_protein	76.09	79.83	77.91	86.52	88.37	87.43
small_molecule	72.77	60.13	65.85	77.07	66.28	71.27
cellular_component	73.57	70.07	71.78	79.30	65.80	71.92
cell_type_or_line	67.60	62.59	65.00	76.85	65.53	70.74
tissue_or_organ	68.34	49.26	57.25	70.58	58.22	63.80
organism_or_species	61.89	65.08	63.44	72.59	75.04	73.79
overall	73.36	71.90	**72.62**	82.25	78.87	**80.53**

[2] These punctuation strings are the ones we used in the submitted model, but have since been revised.

[3] http://www.uniprot.org/help/api

TABLE IV. PRECISION, RECALL, AND F_1 SCORES ACROSS ENTITIES FOR SUBMITTED BIOLOGICAL ENTITY IDENTIFICATION MODEL

Entity types	Strict span match for all annotations			Strict span match for norm. annotations only			Span overlap match for all annotations			Span overlap match for norm. annotations only			Micro-averaged scores for normalized IDs			Macro-averaged scores across captions for normalized IDs		
	P	***R***	***F_1***	***P***	***R***	***F_1***	***P***	***R***	***F_1***	***P***	***R***	***F_1***	***P***	***R***	***F_1***	***P***	***R***	***F_1***
gene_or_protein	50.87	61.31	55.60	52.96	59.61	**56.09**	68.55	82.63	**74.94**	61.51	69.24	65.15	16.98	22.43	19.33	23.24	30.37	16.69
small_molecule	56.23	45.09	**50.04**	65.90	32.22	43.28	68.31	54.78	**60.80**	70.54	34.49	46.33	65.42	39.37	49.16	77.23	47.70	34.11
cellular_component	54.77	43.94	48.76	61.24	41.03	**49.14**	62.89	50.46	**55.99**	65.13	43.64	52.26	54.98	44.97	49.47	67.25	53.98	35.05
cell_type_or_line	65.31	65.03	65.17	82.23	55.15	**66.02**	76.63	76.30	**76.47**	86.58	58.06	69.51	78.42	55.69	65.13	85.58	60.15	55.51
tissue_or_organ	57.24	55.87	**56.55**	61.46	46.67	53.05	67.05	65.44	**66.24**	66.74	50.68	57.61	58.38	44.21	50.32	69.90	54.63	37.92
organism_or_species	74.62	71.52	73.04	85.52	69.50	**76.68**	81.36	77.98	**79.63**	87.82	71.37	78.74	77.23	69.10	72.94	82.48	74.69	65.46

VI. RESULTS AND CONCLUSIONS

A. NER results

Table 3 shows the results of the NERSuite model and the results of the BLSTM-BLSTM-CRF model. Both models were trained on the training and development sets and scores are reported for the test set. The BLSTM-BLSTM-CRF model using word embeddings pretrained on the biomedical domain does significantly better than the NERSuite model across all entity types. Our earlier experiments showed that for certain entity types, the BLSTM-BLSTM-CRF model with randomly initialized word embeddings performs comparably or worse than a CRF model when trained on the BioCreative data. This result emphasizes the significance of word embeddings pretrained on domain-specific data. We are in the process of experimenting with other neural architectures and are seeing promising results. Given that neural models often rely on a large amount of data to generate accurate results, and biomedical NER annotations need to be done by experts, we think distant supervision techniques may be especially helpful. Our manual error analysis indicates that better tokenization schemes might help the model better detect the boundaries of an entity. Also, our model tags entities at the word level, but the task evaluates entities at the byte level, so a model that tags at the byte level may be more suitable.

B. Submission results

In Table 4, we list the scores under various evaluation conditions for the unseen test set. The first four conditions evaluate the NER model, and the last two conditions evaluate grounding tagged entities to identifiers. As expected, the scores are higher when evaluating span overlap entity matches versus strict span entity matches. Interestingly, all entity types have higher F_1 scores when evaluating span overlap match for all annotations versus for normalized annotations only. This indicates that our NER model is better at detecting non-normalized entities across entity types.

In this work, we focus on experimenting with state-of-the-art NER techniques applied to the biomedical domain. We do not spend a comparable effort on grounding techniques, though we are working to improve them. We observe that the grounding method in our submitted model performs the best on *organism_or_species* and *cell_type_or_line*, does ok on *small_molecule*, *cellular_component*, and *tissue_or_organ*, and performs poorly on *gene_or_protein*. One reason the normalization performance of *gene_or_protein* entities is poor is because we use a limited context to ground entities. Similar genes and proteins of difference species often have the same surface forms, and the only way to accurately ground the genes and proteins is to infer the species from the textual context. Another explanation for the poor normalization performance is that genes and proteins have the most variations in surface forms; there are relatively fewer ways to refer to organisms and species, for example. From a manual evaluation of our grounding method, we observe that better organism modeling would help improve the normalization scores. Also, the simple heuristics for segmentation seem to help us more accurately extract short protein and gene entities, but we often make more errors grounding shorter proteins and genes. For future work, we would explore using more contextual evidence to assign entity identifiers.

REFERENCES

[1] Carbon S, Ireland A, Mungall CJ, Shu S, Marshall B, Lewis S, AmiGO Hub, Web Presence Working Group. AmiGO: online access to ontology and annotation data. Bioinformatics. Jan 2009;25[2]:288-9.

[2] Chiu, J. P., & Nichols, E. (2015). Named entity recognition with bidirectional LSTM-CNNs. *arXiv preprint arXiv:1511.08308*.

[3] Cho, H. C., Okazaki, N., Miwa, M., & Tsujii, J. (2010). NERsuite: a named entity recognition toolkit. *Tsujii Laboratory, Department of Information Science, University of Tokyo, Tokyo, Japan*.

[4] Lafferty, J., McCallum, A., & Pereira, F. C. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

[5] Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., & Dyer, C. (2016). Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

[6] Ma, X., & Hovy, E. (2016). End-to-end sequence labeling via bi-directional lstm-cnns-crf. *arXiv preprint arXiv:1603.01354*.

[7] Ramshaw, L. A., & Marcus, M. P. (1999). Text chunking using transformation-based learning. In *Natural language processing using very large corpora* (pp. 157-176). Springer Netherlands.

[8] Sayers E. E-utilities Quick Start. 2008 Dec 12 [Updated 2013 Aug 9]. In: Entrez Programming Utilities Help [Internet]. Bethesda (MD): National Center for Biotechnology Information (US); 2010-. Available from: https://www.ncbi.nlm.nih.gov/books/NBK25500/

SPRENO: A BioC Module for Recognizing and Normalizing Species and Their Model Organisms

A Species recognizer, identify mentioned species name in figure captions

Onkar Singh[1,2], Hong-Jie Dai[3,4]*

[1] Bioinformatics Program, Taiwan International Graduate Program, Institute of Information Science, Academia Sinica, Taipei 115, Taiwan, R.O.C.

[2]Institute of Biomedical Informatics, National Yang-Ming University, Taipei, 112, Taiwan, R.O.C.

[3]Department of Computer Science and Information Engineering, National Taitung University, Taitung, Taiwan, R.O.C.

[4]Interdisciplinary Program of Green and Information Technology, National Taitung University, Taitung, Taiwan, R.O.C.

***Abstract*—**Name entity recognition is a key step in a biomedical text mining task. This becomes more critical and challenging due to the availability of huge amount of biomedical literature. To recognize and identify species become difficult for the domain experts due to the vagueness of the abbreviated term used for model organisms/strains. In this study, we present our species recognition tool–SPRENO (Species Recognition and Normalization) developed for recognizing organism terms mentioned in figure captions. SPRENO is an extension of our previous species recognition tool developed for the BioCreative V BioC task. We developed new algorithms optimized for normalizing organism terms mentioned in figure captions, which consider the contextual information from the corresponding full text. Furthermore, two disambiguation methods are developed to determine the ID of ambiguous organism mentions. One is based on the majority rule to select the ID that has been successfully linked to previously mentioned organism terms. Another is a convolutional neural network model trained by learning the context and the distance information of the target organism mention. We participated the BioCreative VI BioID task and submitted three runs for the assessment of the developed tool. The best micro F-scores achieved by SPRENO on the test set are 0.776 (entity recognition) and 0.755 (entity normalization).

Keywords—Named entity recognition, organism normalization, convolutional neural network

I. INTRODUCTION

The unprecedented growth in biomedical literature necessitates perpetual reformations of automated text mining tools which can correctly extract individual or multiple biomedical entities (e.g. gene/gene products, organism etc.) and transform orderly. However, the complexity of the dynamically changing terminology for the same bio-entity has emerged as a challenging task for the bio-curators. Recognizing bio-entities manually demonstrates high detection accuracy but is time-consuming and labor intensive. It provides us a lot of scope for the researchers to develop automated annotation tools.

The primary task of biomedical text mining is named entity recognition (NER) and normalization of the entity. NER tools are developed to ascertain biomedical entities such as the mentioned species, gene and gene products in biomedical literature. To facilitate downstream tasks, it is very important to accurately recognize those entities and associate them with their corresponding database/ontology IDs [1]. As one of the participants in the BioCreative VI Bio-ID assignment track, we extended our previous species recognition tool [2] for recognizing organism terms mentioned in figure captions. Comparing with the recognition of species terms mentioned in abstracts or full texts, which have been studied in previous works [3, 4], the process of recognizing terms described in figure captions is challenging owing to the absence of specific criterion, unique terminology, and unexplained abbreviated words. The ambiguous nature of the abbreviated terms requires a strategy to process full text to find the full term which makes exceedingly tough for the domain expert to identify organisms and link them to their unique taxonomy IDs. For example, the abbreviated term SIN may stand for Sindbis virus or the same term in gene ontology referring to Sex-lethal interactor gene (*Drosophila melanogaster*).

Another challenge that was encountered during the recognition of organism terms in the BioID task was the identification of strains. Authors use specialized terms of strains/model in figure captions to describe their experimental observations. For example, the terms of inbred strains of the mouse include C57BL/6J, R6/2, DBA/2J etc. Aside from the organism tagger developed by Naderi, Kappler, Baker and Witte [5], those strain mentions cannot be recognized by most of the current openly available tools.

In this paper, we present our new species recognition tool, SPRENO (Species Recognition and Normalization). We extended the lexicon used by our previous species recognition tool by including organism terms and common terms used to refer to strains or models. We also developed new algorithms optimized for normalizing organism terms mentioned in figure captions, which consider the contextual information from the corresponding full text. Finally, disambiguation methods based on the majority rule and the convolutional neural network (CNN) were developed to determine the ID of ambiguous organism mentions.

II. METHOD

A. Lexicon Extended with Terms of Strain/model

As mentioned in the previous section, the lexicon used by our previous species recognition tool only includes species terms, as well as prefixes in a gene name, which can refer to the species. At present work we extended the lexicon by adding the terms used for organisms, common terms and abbreviations used for strains/model organism. Table 1 summarizes the resources used in this work.

TABLE I. RESOURCES FOR STRAIN/MODEL TERMS

Source	Organism
http://www.informatics.jax.org/inbred_strains/mouse/STRAINS.shtml	Mouse
http://www.criver.com/find-a-model	Mouse
http://gcm.wfcc.info/speciesPage.jsp?strain_name=Lactobacillus%20acidophilus#specTopgcm.wfcc.info	Lactobacillus acidophilus
https://gold.jgi.doe.gov/organisms?Organism.Domain=BACTERIAL&Organism.Type%20Strain=Yes&Organism.Active=Yes	Bacteria
https://byo.com/resources/yeast	Yeast

B. Normalization and Disambiguation Approach

Fig. 1 demonstrates the workflow of the developed organism recognizer. We extended our previous BioC library[1] to support the process of figure captions represented in the Bio C format defined by the Bio-ID task. The developed library is then used to load the figure captions and their corresponding full-text article. Although the goal of Bio-ID task is to identify the organism terms in figure captions only, we still process both the full text and figure captions. Therefore both the full text and figure captions are preprocessed to detect sentence boundaries, tokens, part-of-speech (PoS) tags and full name-abbreviation pairs.

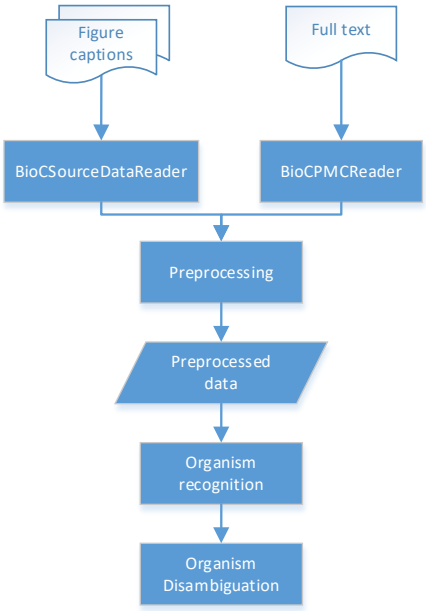


Fig. 1. Workflow of the developed organism recognizer.

When processing the full text, the base forms of the full names listed as the full name-abbreviation pairs found from the entire full-text are matched with our lexicon before performing the actual organism recognition process. If the full name is considered to be an organism term, both the name and its abbreviation are added to the organism lexicon for the one-off matching of the given full-text. Otherwise, the pairs are blacklisted for the current full-text. We then process the text of figure captions based on the updated lexicon. Through this way, we can reduce the ambiguity of abbreviated terms frequently used in figure captions. Finally, the algorithm developed in our previous work [2] was employed to recognize all organism terms from the full text by exploiting linguistic information and match against the extended dictionary. After identifying all of the organism candidates, the PoS information is used to filter out false-positive cases such as candidates with PoS as a verb.

In order to reduce the ambiguity of the recognized organisms in figure captions, we applied two disambiguation methods. The first is a rule-based approach which uses the normalization information from full text. The algorithm follows the similar idea of our multi-stage normalization algorithm [6] to disambiguate ambiguous organism terms by exploiting the normalization information from the entire article. The majority rule is used to select the ID that has been successfully linked to previously mentioned organism terms.

The second disambiguation method is a machine learning based method based on CNN. We formulated the disambiguation problem as a binary classification task and generated the training set based on the outcome of our organism recognizer. The generated training set includes both, the normalized terms and ambiguous terms along with the candidate IDs as well as their context in figure captions. Fig. 2 shows an overview of the developed CNN model for organism disambiguation.

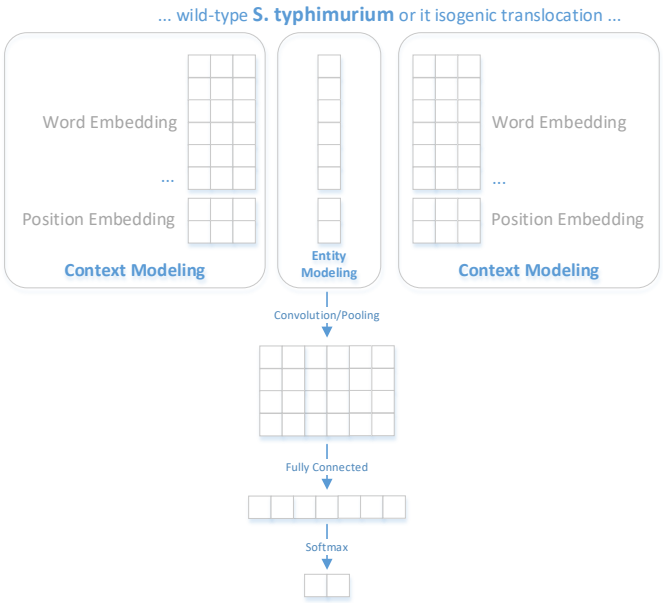


Fig. 2. The developed neural network model for organism disambiguation.

[1] https://www.nuget.org/packages/NTTU.BigODM.Bio.BioC/

The input of our model includes the context of the mention and the mention's candidate record from the NCBI taxonomy database. The output of our CNN is the probability distribution over two possible outcomes {yes, no}. The context is represented by word representations and the distance between a context word and the target mention based on the consideration that a closer context word might be more informative than a farther one for disambiguating that mention [7]. For the entity modeling, the target entity was represented as the official symbol recorded in the taxonomy database.

The pre-trained PubMed word embedding released by Moen and Ananiadou [8] was used for representing the words. Therefore, the dimension of word vector was set as 200. The learning rate and the window size of CNN were empirically set as 0.01 and 2, respectively.

C. Common Term Recognition and Normalization

We observed that in the Bio-ID corpus, the annotators tend to annotate common terms like larvae and embryos for organisms such as *Drosophila melanogaster* and *Danio rerio* (zebrafish) depending on the context mentioned in the full-text article or figure captions. For example, in Figure 1, the term "embryo" indicates the species zebrafish. We analyzed the training set of Bio-ID corpus to collect all possible grounding IDs for a given common term.

> (A) Yolk opaqueness and LC3 puncta formation in spns1-mutant **zebrafish embryos**. For EGFP-LC3 transgenic spns1-mutant [Tg(CMV:EGFP-LC3);… and spns1 mutant (spns1−/−) (lower) **embryos** at 84 hpf are shown.

Fig. 3. An example text indicates the common term (embryos) and its related species (zebrafish).

When processing a given article, if our tool detects an occurrence of the considered common terms, an algorithm was developed to select an ID from the term's possible IDs. The selection strategy is designed as a way to select the ID which appeared most frequently in the entire full text.

III. Results

We submitted three runs to assess the performance of the developed organism recognizer module. At first run, the rule-based normalization method utilizing full-text information only was applied. We set a threshold at two to filter those organism names assigned for more than two IDs after the disambiguation process. In the second run, the threshold for filtering increased to 10, i.e. those organism mentions having more than 10 IDs were filtered out. After that, we employed the developed CNN model to select the ID with the highest likelihood. At last, we performed third run where we used the original lexicon from our previous work along with the CNN-based disambiguation. The threshold for the last run was set to infinite.

Table II shows the recognition results for organism on the test set of the Bio-ID task. Two matching criteria are used. The strict span condition depicts that annotator annotates the term "zebrafish" and another annotator annotates "spns1-mutant zebrafish" then the match function will fail to identify. While in overlap situation the term is considered to be a match.

As shown in Table II, the first run achieves the best recall (R) and F-score among others. The second run with the developed CNN filter achieves better precision (P) under both matching criteria. However, as we increase the threshold the performance declining gradually.

TABLE II. Official Recognition Results on the Test Set

No of Run	NER			
	Criterion	**Precision**	**Recall**	**F-measure**
Run 1	Strict	0.662	0.873	0.753
	Overlap	0.681	0.910	0.776
Run 2	Strict	0.670	0.730	0.699
	Overlap	0.689	0.751	0.719
Run 3	Strict	0.516	0.250	0.337
	Overlap	0.535	0.260	0.350

In order to describe the normalization result, we used two methods to get average statistical scores i.e. micro average method and macro average method. Table III shows the official results on the test set. Comparing the normalization results with the recognition results we can observe that the developed CNN disambiguation method can improve the precision of the two tasks but reduce the recall. We believe that it may due to that the corpus released by the Bio-ID task does not exhaustively annotate all organism terms mentioned in figure captions.

TABLE III. Official Normalization Results on the Test Set

Run	Micro-P	Micro-R	Micro-F-score	Macro-P	Macro-R	Macro-F-score
1	0.660	0.882	0.755	0.709	0.924	0.685
2	0.668	0.760	0.711	0.732	0.824	0.611
3	0.525	0.327	0.403	0.766	0.462	0.273

IV. Conclusion

In summary, we introduce our new species recognition tool SPRENO which can recognize organism terms mentioned in figure captions. The developed tool will be released on https://www.nuget.org/packages/NTTU.BigODM.Bio.NER.Species/.

Acknowledgment

This work was supported by the Ministry of Science and Technology of Taiwan [grant numbers MOST-105-2221-E-143-003 and MOST-106-2221-E-143-007-MY3].

References

[1] A. Akkasi, and E. Varoglu, "Improving Biochemical Named Entity Recognition Performance Using PSO Classifier Selection and Bayesian Combination Method," *IEEE/ACM Trans Comput Biol Bioinform*, May 18, 2016.

[2] H.-J. Dai, O. Singh, J. Jonnagaddala, and E. C.-Y. Su, "NTTMUNSW BioC modules for recognizing and normalizing species and gene/protein mentions," *Database,* vol. 2016, January 1, 2016, 2016.

[3] S. Kim, R. Islamaj Doğan, A. Chatr-Aryamontri, C. S. Chang, R. Oughtred, J. Rust, R. Batista-Navarro, J. Carter, S. Ananiadou, S. Matos, A. Santos, D. Campos, J. L. Oliveira, O. Singh, J. Jonnagaddala, H.-J. Dai, E. C.-Y. Su, Y.-C. Chang, Y.-C. Su, C.-H. Chu, C. C. Chen, W.-L. Hsu, Y. Peng, C. Arighi, C. H. Wu, K. Vijay-Shanker, F. Aydın, Z. M. Hüsünbeyi, A. Özgür, S.-Y. Shin, D. Kwon, K. Dolinski, M. Tyers, W. J. Wilbur, and D. C. Comeau, "BioCreative V BioC track overview: collaborative biocurator assistant task for BioGRID," *Database: The Journal of Biological Databases and Curation,* vol. 2016, pp. baw121, 09/01

[4] E. Pafilis, S. P. Frankild, L. Fanini, S. Faulwetter, C. Pavloudi, A. Vasileiadou, C. Arvanitidis, and L. J. Jensen, "The SPECIES and ORGANISMS Resources for Fast and Accurate Identification of Taxonomic Names in Text," *PLoS ONE,* vol. 8, no. 6, pp. e65390, 2013.

[5] N. Naderi, T. Kappler, C. J. Baker, and R. Witte, "OrganismTagger: detection, normalization and grounding of organism entities in biomedical documents," *Bioinformatics,* vol. 27, no. 19, pp. 2721-9, Oct 1, 2011.

[6] H.-J. Dai, P.-T. Lai, and R. T.-H. Tsai, "Multistage Gene Normalization and SVM-Based Ranking for Protein Interactor Extraction in Full-Text Articles," *IEEE/ACM TRANSACTIONS ON COMPUTATIONAL BIOLOGY AND BIOINFORMATICS,* vol. 7, no. 3, pp. 412-420, 2010.

[7] Y. Sun, L. Lin, D. Tang, N. Yang, Z. Ji, and X. Wang, "Modeling Mention, Context and Entity with Neural Networks for Entity Disambiguation." pp. 1333-1339.

[8] S. Moen, and T. S. S. Ananiadou, "Distributional semantics resources for biomedical text processing," LBM, 2013.

TurkuNLP Entry for Interactive Bio-ID Assignment

Suwisa Kaewphan[1,2,3], Farrokh Mehryary[2,3], Kai Hakala[2,3], Tapio Salakoski[1,2] and Filip Ginter[2]
1. Turku Centre for Computer Science, Turku, Finland
2. Department of Future Technologies, University of Turku, Turku, Finland
3. University of Turku Graduate School, University of Turku, Turku, Finland

***Abstract*—We participate in BioCreative VI: Interactive Bio-ID Assignment (Bio-ID) track by developing systems capable of named entity recognition and normalization of 6 entity types, namely *Protein, Cell, Organism, Tissue, Molecule and Cellular*. Our named entity recognition system is based on conditional random fields. For named entity normalization, we apply fuzzy matching and rule-based system to disambiguate and assign unique identifiers to the entities. The official evaluation shows that average F1-scores of all entity types for our recognition and normalization systems on strict span offsets are 0.720 and 0.668, respectively.**

Keywords—CRF; fuzzy matching; rule-based; Simstring

I. Introduction

The main goal of BioCreative VI (Bio-ID) track is to annotate text with the entity types and IDs for organism, gene, protein, miRNA, small molecules, cellular components, cell types and cell lines, tissues and organs, in order to facilitate the curation process. The task principally consists of two major subtasks: i) named entity recognition (NER) and ii) named entity normalization (NEN).

On one hand, several machine learning-based approaches, such as support vector machines and neural networks, have been applied to NER tasks with varying entities ranging from genes to diseases, chemicals and anatomical parts [1,2]. The most recent successful approaches include conditional random field (CRF) classifiers and neural networks [1,3,4]. The approaches for NEN, on the other hand, are largely based on string edit distance and TFIDF weighted vector space representations with a variety of preprocessing approaches to remove the written variations [5,6].

Our system, capable of recognizing all six types of entities and assigning the corresponding identifiers, is based on CRF classifiers, fuzzy matching and a rule-based system. In the following sections we describe our system and its performance based on the official evaluation for both recognition and normalization tasks.

II. Methods

A. Preprocessing

We preprocess the documents by using the publicly available tool [77] converting the character encodings to ASCII. The characters with the missing mapping, such as smiley faces and calendar symbols, are thus replaced with '-' (dash). Subsequently we split the documents into sentences and further tokenize and part-of-speech (POS) tag them using GENIA sentence splitter [8], NERsuite tokenizer and NERsuite POS tagger modules [9], respectively.

Some of the documents contain incorrect word boundaries such as 'mouseliverlysosomes' which should have been written as 'mouse liver lysosome'. While the result of tokenization is overall satisfactory, it is incapable of correctly splitting these words into tokens. We thus resolve this by additional tokenization using the known tokens from the corresponding full-text document. Specifically, we split the tokens using the span of the longest matching document tokens. To reduce the chance of mistakenly tokenizing correct tokens, we only re-tokenize the tokens that belong to noun phrases. Finally, we re-apply POS tagging to complete the data preprocessing.

B. Ontology and controlled vocabularies

We prepare a set of controlled vocabularies and ontologies to assist named entity recognition and normalization. List of concept names and ontologies we used include ChEBI [10] and PubChem [11] (for *Molecule*), Entrez Gene [12] and Uniprot [13] (for *Protein*), NCBI Taxonomy [14] (for *Organism)*, Uberon [15] (for *Tissue*), Cellular Component Ontology [16] (for *Cellular* component) and Cell Ontology (http://purl.obolibrary.org/obo/uberon.owl) and Cellosaurus (http://web.expasy.org/cellosaurus) (for *Cell)*. We preprocess the lists by removing non-alphanumeric characters and lowercasing the symbols.

Specifically for NCBI Taxonomy, we additionally expand the ontology by adding the commonly used abbreviations for scientific names. For binomial nomenclature of names in species rank, we abbreviate the genus while the rest of the names such as species epithet, varieties, strains and substrains, remain the same. For example, 'Escherichia coli O.1197' is abbreviated as 'E. coli O.1197', 'E coli O.1197', 'Es. coli O.1197' and 'Es coli O.1197'. This rule applies to all organisms, except for scientific names of organisms in Viruses and Viroids superkingdoms, since the scientific names do not usually follow binomial nomenclature but are in the form of [Disease] virus [17]. Acronyms are often used as abbreviated scientific names for viruses, for example ZYMV is the acronym of Zucchini yellow mosaic virus, and thus we also add acronyms to the ontology.

C. Named Entity Recognition

For the given training data, we first completely remove annotations for *Assay* entity type and combine *miRNA* and *Gene* with *Protein* annotations. Hence, the total entity counts are 58476, 7476, 6312, 11213, 10604 and 7888 for *Protein*, *Cellular*, *Tissue*, *Molecule*, *Cell* and *Organism*. We then randomly partition the training data into a training and a

development set, containing 455 and 115 documents respectively.

We train our NER system on the training set using the NERsuite (http://nersuite.nlplab.org/)—a named entity recognition toolkit—and optimize it against our development set. We train a single CRF model capable of detecting all possible entity types and use micro-averaged F1-score as the optimization metric, derived from the official evaluation script. To achieve higher performance in NER, we directly provide NERsuite with dictionaries through dictionary-tagging module with no further preprocessing or normalization. We compare the performance of different dictionaries on development data using default NERsuite hyperparameters.

For final prediction of the test set, we merge the training and the development sets and re-train the CRF on this data using the best found hyperparameters.

D. Named Entity Normalization and Disambiguation

Our normalization approach is primarily based on fuzzy string matching algorithm where both entity and ontology terms are converted to vectors using character n-gram frequencies. Cosine similarity is then used for calculating similarity between detected entity and ontology terms. In this study, we use Simstring [18], a library for approximate string matching, to retrieve the ontology terms with highest cosine similarity with queried entity, regardless of the type of the terms.

The tagged entities resulting from the NER system are preprocessed using the same approaches we use on dictionaries and ontologies, by removing the punctuations and lowercasing, as described previously. We utilizes approximate string matching approach to all entity types except for *Protein*, which we instead apply 'exact string matching' to retrieve matching identifiers.

Some of the ontology terms are not uniquely linked to a single identifier, but multiple ones. For *Cell*, *Cellular*, *Molecule* and *Tissue*, a random identifier is selected. The selected random identifier is subsequently applied throughout the document. For *Organism* and *Protein*, we develop two separate rule-based systems to uniquely assign an identifier.

For *Organism*, we use taxonomy tree and the following disambiguation rules to assign a taxon identifier to *Organism*. These rules are sequentially applied if the previous rule results in more than one identifier.

1. Take identifier with highest cosine similarity score and its taxonomic rank is under species, which also includes subspecies, strain, variety and no rank.
2. Take identifier of previous mentioned *Organism* if abbreviations match.
3. Take identifier of previous mentioned *Organism* if acronyms match.
4. Take identifier of previous mentioned *Organism* of the same genus.
5. Take identifier of a model organism of the same genus.
6. Take identifier of the most studied organism in PubMed-Central Open Access section.
7. Take a random identifier.

Protein contain the most ambiguous names as the same protein names can be found in multiple organisms if they have the same function or shared sequence identity [19]. Therefore, the information about the *Organism* is crucial for *Protein* normalization. We therefore employ the results of our *Organism* normalization system and use the taxon identifiers to disambiguate *Protein*. However, multiple taxon identifiers can be recognized in a single document, hence we adapt rule-based system proposed by [1] to generate candidate taxon identifiers for the *Protein*. The list of candidate taxon identifiers are ordered according to the following rules.

1. *Organism* mentioned inside *Protein* text span
2. *Organism* mentioned before *Protein* within the same sentence
3. *Organism* mentioned after *Protein* within the same sentence
4. *Organism* mentioned in the previous caption
5. *Organism* mentioned in the same document

In addition, we perform query expansion to generate candidate *Protein* names to cover potential Uniprot and Entrez Gene symbol variations by using stripping algorithm [20]. The algorithm recursively removes common words, such as protein, gene and RNA, and *Organism* from *Protein* to produce a canonical form which includes minimal symbols that are gene symbols in the Entrez Gene database. For instance, 'p53 protein' will result to 'p53'. Finally, the canonical forms are subsequently lower-cased and punctuation-removed. The list of candidate *Protein* names are then ordered by the string length.

For each taxon identifier, we use 'exact string matching' to retrieve corresponding *Protein* identifier. The search starts with the longest candidate *Protein* name and stops when the identifier is found. In case of multiple identifiers, a random one is selected.

III. Result and Discussion

E. Name Entity Recognition

Incorrect word boundaries can result in multiple types of entity annotations for a given token. For example, 'mouseskinfibroblasts' contains the annotations for *Organism*, *Tissue* and *Cell*. Since we train a single CRF-based model to recognize all types of entities, having one token representing multiple entities would have caused the loss of training examples as NERsuite does not support multilabel classification. As mentioned in Method section, we resolve this issue by re-tokenizing the tokens using known tokens from the provided full-text document. The result for recovering the training examples is significant as tokenization from NERsuite alone yields about 97% of the annotations, while this step increases the number of annotations by additional 2pp, equivalent to more than 2000 annotations. As a result, we recover more than 99% of the original annotations with *Organism* with the highest increase in coverage.

TABLE I. COMPARISON OF ANNOTATION COUNTS BETWEEN TOKENIZATION APPROACHES

re-tokenization	*Prot*	*Cellu*	*Tiss*	*Mole*	*Cel*	*Org*
without	97.178	99.772	95.951	96.107	97.099	93.691
with	99.187	99.866	99.842	99.424	99.559	98.921

a. The comparison of annotation counts between preprocessing with only NERsuite tokenization module (without) and with both NERsuite tokenization and additional tokenization (with). The numbers are percents of annotations compared to the provided data presented for each entity type.

It has been demonstrated that domain knowledge, such as controlled vocabularies, is important to attain good performing NER model [3,4]. In this study, we use dictionaries to add features for classifier and compare the model performance on the development data. As shown in Table II, there is no clear performance improvement when adding dictionary features in either strict or overlap modes of evaluation. In the case of cellular component from GO, the performance of NER is however, lower than other models by more than 6pp in F-measure. As a result, we train our model without using any additional dictionary features.

TABLE II. OFFICIAL EVALUATION OF NER SYSTEM ON DEVELOPMENT DATA

Dictionary/ Ontology	**Precision / Recall / F-measure**	
	Strict	***Overlap***
Uberon	**0.787 / 0.688 / 0.734**	**0.882 / 0.771 / 0.823**
ChEBI	0.763 / 0.689 / 0.724	0.865 / 0.780 / 0.821
GO	0.652 / 0.687 / 0.669	0.789 / 0.830 / 0.809
Cellosaurus	0.780 / 0.687 / 0.730	0.875 / 0.772 / 0.820
NCBI Taxonomy	**0.785 / 0.689 / 0.734**	**0.880 / 0.772 / 0.823**
NCBI Gene	0.770 / 0.688 / 0.727	0.870 / 0.778 / 0.821
Cell ontology	**0.788 / 0.687 / 0.734**	**0.883 / 0.770 / 0.823**
No dictionary	0.788 / 0.686 / 0.734	0.882 / 0.769 / 0.822

We finally train NERsuite model on combined training and development sets. The resulting model is subsequently used for tagging the entities in the test dataset. The official evaluation results, shown in Table III, demonstrate that our NER system performs best on *Organism*, achieving F-measure of 0.834. The performance of the system is moderate for *Cell* and *Protein* with F-measure of 0.743 and 0.734, respectively. For the other three entity types, *Tissue*, *Cellular* and *Molecule*, our system shows comparatively lower performances with F-measure of 0.668, 0.642 and 0.579, respectively. *Cellular* proves to be the most difficult entity to recognize. Overall, the performance of model is moderate across all entity types, achieving F-measure of 0.720 and 0.790 on strict and overlap evaluation criteria.

TABLE III. OFFICIAL EVALUATION OF NER SYSTEM ON TEST DATA

Entity	**Precision / Recall / F-measure**	
	Strict	***Overlap***
Cell	0.783 / 0.708 / 0.743	0.841 / 0.760 / 0.799
Cellular	**0.673 / 0.508 / 0.579**	**0.728 / 0.550 / 0.627**
Protein	0.729 / 0.739 / 0.734	0.825 / 0.836 / 0.831
Organism	**0.860 / 0.809 / 0.834**	**0.878 / 0.826 / 0.852**
Molecule	0.775 / 0.587 / 0.668	0.796 / 0.603 / 0.686
Tissue	0.727 / 0.575 / 0.642	0.793 / 0.627 / 0.701
All	**0.747 / 0.694 / 0.720**	**0.821 / 0.762 / 0.790**

F. Name Entity Normalization and Disambiguation

The performance of normalization system is heavily depending on the NER system performance since unrecognized and incorrect spans entities are automatically classified as false negative and false positives, respectively. We thus evaluate our normalization system on the development set based on the gold standard entity mentions to compare the different approaches on different entity types.

TABLE IV. OFFICIAL EVALUATION OF NEN SYSTEM ON DEVELOPMENT DATA

Entity	**Precision / Recall / F-measure**	
	Strict	***Overlap***
Cell	0.902 / 0.946 / 0.923	0.935 / 0.980 / 0.957
Cellular	**0.974 / 0.929 / 0.951**	**0.980 / 0.934 / 0.957**
Protein	0.878 / 0.591 / 0.706	0.902 / 0.606 / 0.725
Organism	0.977 / 0.887 / 0.930	0.993 / 0.901 / 0.945
Molecule	**0.963 / 0.488 / 0.647**	**0.969 / 0.491 / 0.651**
Tissue	0.920 / 0.978 / 0.948	0.930 / 0.988 / 0.958
All	0.914 / 0.700 / 0.793	0.933 / 0.716 / 0.810

Our normalization system performs relatively well on *Cell*, *Cellular*, *Organism* and *Tissue*, where the F-measure ranges from 0.923 to 0.951 under strict criteria. However, the performance of the system drops dramatically on *Molecule* and *Protein*, as their recall of both entities are significantly lower than their precision counterpart. For *Protein*, the exact string matching and a set of taxon identifiers are probably attributing factors for a low recall as these two criteria are probably too stringent resulting in almost half of the *Protein* not being linked to an associated identifier. For *Molecule*, the lower recall is most likely caused by some other factor since the approximate pattern matching was used.

When evaluated against test set, the normalization results differ from gold standard development data as the overall performance is largely depending on the NER system output. As shown in Table V, the normalization performance does not appear to drop drastically even when applied on predicted entities instead of the gold standard mentions.

TABLE V. OFFICIAL EVALUATION OF NEN SYSTEM ON TEST DATA

Entity	Precision / Recall / F-measure	
	Strict	***Overlap***
Cell	0.774 / 0.699 / 0.735	0.830 / 0.750 / 0.788
Cellular	**0.685 / 0.482 / 0.566**	**0.737 / 0.519 / 0.609**
Protein	0.795 / 0.543 / 0.645	0.823 / 0.561 / 0.667
Organism	**0.857 / 0.797 / 0.826**	**0.877 / 0.815 / 0.845**
Molecule	0.787 / 0.581 / 0.668	0.803 / 0.593 / 0.682
Tissue	0.604 / 0.542 / 0.572	0.669 / 0.600 / 0.632
All	0.775 / 0.586 / 0.668	0.809 / 0.612 / 0.697

IV. CONCLUSIONS AND FUTURE WORK

We approach BioCreative Bio-ID task by training a single CRF-based model to recognize all entity types and we link them to their corresponding database identifiers using approximate pattern matching algorithm. For *Protein* and *Organism*, we utilize the ontology structure and surrounding context to disambiguate the entities with multiple identifier candidates. Our systems, evaluated independently, demonstrate a moderate performance overall. However, a lower performance for most types of entities is observed when recognition and normalization are evaluated jointly as the F-score is largely determined by the F-score of the recognition system.

CRF-based classifiers have been a relatively successful tool for entity recognition in biomedical domain, demonstrating state-of-the-art for several entity types. However, it has been recently shown that neural networks with only word embeddings as features can outperform traditional CRF-based NER systems with manually crafted features [1]. Thus, our future work includes developing a neural network-based NER system capable of recognizing multiple types of entities.

Our normalization system for *Protein* and *Molecule* demonstrate a lagging performance when compared with other entities. For *Protein*, applying relaxed string matching in addition to improving the organism assignment algorithm can potentially improve the performance . For *Molecule*, our future work lies on identifying contributing factors that lower the recall and adjusting the system accordingly.

Our current normalization system is limited and time-consuming as it applies several manually generated rules which do not generalize to normalizing other entity types. Thus developing a machine learning system that can be trained on the annotations of new entity type would be an ideal solution for the normalization task. Since the conventions of naming biomedical entities vary among entity types, a unified normalization system can be a challenging task.

ACKNOWLEDGMENT

Computational resources are provided by CSC-IT Center For Science Ltd., Espoo, Finland. This work is supported by ATT Tieto käyttöön grant.

REFERENCES

1. Ding, R., Arighi, C.N., Lee, J.Y., Wu, C.H. and Vijay-Shanker, K. (2015) pGenN, a gene normalization tool for plant genes and proteins in scientific literature. PloS one, 10(8), p.e0135305.

2. Habibi, M., Weber, L., Neves, M., Wiegandt, D.L. and Leser, U. (2017) Deep learning with word embeddings improves biomedical named entity recognition. *Bioinformatics*, 33(14), pp.i37-i48.

3. Kaewphan, S., Van Landeghem, S., Ohta, T., Van de Peer, Y., Ginter, F. and Pyysalo, S. (2015) Cell line name recognition in support of the identification of synthetic lethality in cancer from text. *Bioinformatics*, 32(2), pp.276-282.

4. Pyysalo, S. and Ananiadou, S. (2014) Anatomical entity mention recognition at literature scale. *Bioinformatics*, 30(6), pp.868-875.

5. Mehryary, F., Hakala, K., Kaewphan, S., Björne, J., Salakoski, T. and Ginter, F. (2017) End-to-End System for Bacteria Habitat Extraction. In *Proceedings of BioNLP 2017*, pp.80-90.

6. Wei, C.H., Kao, H.Y. and Lu, Z. (2015) GNormPlus: An Integrative Approach for Tagging Genes. Gene Families, and Protein Domains, *BioMed Research International*, 918710.

7. Pyysalo, S., Ginter, F., Moen, H., Salakoski, T. and Ananiadou, S. (2013) Distributional Semantics Resources for Biomedical Text Processing. In *Proceedings of LBM 2013.* pp. 39-44.

8. Sætre, R., Yoshida, K., Yakushiji, A., Miyao, Y., Matsubayashi, Y. and Ohta, T. (2007) AKANE system: protein-protein interaction pairs in BioCreAtIvE2 challenge, PPI-IPS subtask. *In Proceedings of the second biocreative challenge workshop*, Vol. 209, p. 212.

9. Tsuruoka, Y., Tateishi, Y., Kim, J.D., Ohta, T., McNaught, J., Ananiadou, S. and Tsujii, J.I. (2005) Developing a robust part-of-speech tagger for biomedical text. In *Panhellenic Conference on Informatics*, pp. 382-392, Springer, Berlin, Heidelberg.

10. Degtyarenko, K., De Matos, P., Ennis, M., Hastings, J., Zbinden, M., McNaught, A., Alcántara, R., Darsow, M., Guedj, M. and Ashburner, M. (2007) ChEBI: a database and ontology for chemical entities of biological interest. *Nucleic acids research*, 36(suppl_1), pp.D344-D350.

11. Bolton, E.E., Wang, Y., Thiessen, P.A. and Bryant, S.H. (2008) PubChem: integrated platform of small molecules and biological activities. *Annual reports in computational chemistry*, 4, pp.217-241.

12. Maglott, D., Ostell, J., Pruitt, K.D. and Tatusova, T. (2010) Entrez Gene: gene-centered information at NCBI. *Nucleic acids research*, 39(suppl_1), pp.D52-D57.

13. UniProt Consortium. (2014) UniProt: a hub for protein information. *Nucleic acids research*, p.gku989.

14. Federhen, S. (2011) The NCBI taxonomy database. *Nucleic acids research*, 40(D1), pp.D136-D143.

15. Mungall, C.J., Torniai, C., Gkoutos, G.V., Lewis, S.E. and Haendel, M.A. (2012) Uberon, an integrative multi-species anatomy ontology. *Genome biology*, 13(1), p.R5.

16. Gene Ontology Consortium. (2004) The Gene Ontology (GO) database and informatics resource. *Nucleic acids research*, 32(suppl 1), pp.D258-D261.

17.. Fauquet, C.M. and Pringle, C.R. (1999) Abbreviations for invertebrate virus species names. *Archives of virology*, 144(11), pp.2265-2271

18. Okazaki, N. and Tsujii, J.I. (2010) August. Simple and efficient algorithm for approximate dictionary matching. In *Proceedings of the 23rd International Conference on Computational Linguistics*. pp.851-859. Association for Computational Linguistics.

19. Chen, L., Liu, H. and Friedman, C. (2004) Gene name ambiguity of eukaryotic nomenclatures. *Bioinformatics*, 21(2), pp.248-256.

20. Van Landeghem, S., Ginter, F., Van de Peer, Y. and Salakoski, T. (2011) EVEX: a PubMed-scale resource for homology-based generalization of text mining predictions. *In Proceedings of BioNLP 2011 workshop*, pp.28-37. Association for Computational Linguistics.

Organism named entity recognition on an enhanced CRF-based model and normalization for BioCreative VI bioentity normalization task

Fu-Chien Yang [1], Shiang-Chi Hsu [1], Hung-Yu Kao [1,2]
Institute of Medical Informatics, National Cheng Kung University, Tainan, Taiwan, R.O.C. 701[1]
Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan, Taiwan, R.O.C. 701[2]

***Abstract*—Bioconcept curation is important to improve research on bio-literatures. Among bioconcepts, organism named entity recognition has been studied for a long time but without much success. According to this, we decided to develop a system for curating organisms and normalizing the entities of figure captions. In this work, we split the system in two parts, one is named entity recognition, and the other is entity normalization. In the first part, we leverage conditional random fields (CRFs) with several linguistic features that assist recognition, and in the second part, we utilize some heuristic ways to enhance the ability to identify the taxonomy ID for each entity. At the bioentity normalization task of BioCreative VI Bio-ID task, our system obtained 0.81 precision in the recognition stage and 0.61 precision in the normalization stage.**

Keywords—conditional random fields; taxonomy named entity recognition and normalization; Biomedical text mining

I. INTRODUCTION

In the biomedical field, biomedical publications are rapidly increasing. Text mining technology has become an integral part for biomedical literatures. Through natural language processing (NLP) techniques analysis, curating useful information within corpora becomes more accurate and lets us readily find relevant information. If there is too much irrelevant information in the corpus, existing techniques will struggle. As biomedical literature grows, so do the number of bioconcepts that require curation – e.g. chemicals, diseases, genes and organisms. There are many existing outstanding tools for curating bioconcepts, such as AuDis[1], OrganismTagger[2] and TaggerOne[3] etc. – but few address the need for curation of organisms. Therefore, we chose to participate in the BioCreative VI Task 1: Interactive Bio-ID Assignment, and we chose the bioentity normalization task to recognize organism named entities and normalize.

This task consists of a collection of figure captions from PubMed which are curated by the SourceData team. The format of the training data is BioC format, but to be easier to use in our system, we transform every caption into Pubtator format, and in the end of our system, we turn our result back into BioC format. After observing the captions in the training data, we found out that there are some words that are different from the original article, such as the word 'Sykfl/flmice' in corpus, but in original article it should be 'Sykfl/fl mice' that should have a space inside the word, and we developed some ways to overcome this problem. We also leveraged useful features and methods from other systems in our system, such as SR4GN[4] and AuDis[1]. Additionally, abbreviations of species names are widespread as well as the use of common English names instead of Latin names, which are easy to read but will cause difficulties for taxonomic identification of the organisms described in the captions. The use of acronyms, which can be both species specific and species independent, also poses challenges for recognition tasks. Lastly, incorrect spelling has created more ambiguity.

In this paper, in section II we will introduce the systems referenced; in section III we explain the method we use for developing our system; in section IV we will report the results of our experiments and discuss the differences between each experiment; finally, in section V we state our conclusions and future work.

II. RELATED WORK

SR4GN[4] identifies and disambiguates gene names, it also focus on species detection and recognition. The method to identify the main species mentioned in the article has been integrated into our system. The method gives more weight to species when the species mention occurs in the title as compared to the abstract. In particular, it will double counts on the frequency of the species mentions in the title. However, when multiple species have the same number of occurrences in a document, the author adopted a tie-breaking strategy that uses the global frequency of different species in the Linnaeus corpus. Another outstanding system is AuDis[1], which is for disease name entity recognition and normalization, but the method and features that the author used inspired us to develop this work.

Linnaeus[5] uses a dictionary-based approach to recognize species names and develops a set of heuristics to resolve ambiguous mentions. As a standalone and open source tool,

Linnaeus has been widely used in many the biomedical literature. In LINNAEUS-species-corpus, it performs with 94% recall and 97% precision.

OrganismTagger[2] (OT) is another well-known system for curating organisms, it is a hybrid rule-based/machine learning system to extract organism mentions from the biomedical literature. OT addresses some challenges and makes some contributions, such as, a machine learning-based classifier for strain detection, and tools for automatically generating lexical and ontological resources from a copy of the NCBI Taxonomy database which allow the system to be updated by the users. On their manually annotated OT corpus, the OT achieves a precision of 95%.

In addition to the aforementioned systems, more recently, there is a new detection and recognition technology called TaggerOne[3], which is currently the state-of-the-art technology for tagging bioconcepts. The system is the first machine learning model for joint named entity recognition (NER) and normalization using semi-Markov models during both training and prediction. The result of NER F-measure of TaggerOne[3] in NCBI Disease corpus is 0.829 and in BioCreative V CDR corpus is 0.914.

Conditional Random Fields (CRFs)[6] is a popular probabilistic method for structured prediction. While most classifiers predict a label with just a single sample without considering other neighboring samples, CRF can take context into account which makes it perform better on predicting tokens' labels. For instance, the linear chain CRF which is famous in natural language processing and is used in our system, predicts sequences of labels for sequences of input samples.

III. METHODS

To deal with the organism identification problem, we designed a semantic based recognition system which includes two modules as shown in Fig. 1. First, taxonomy named entity recognition. We utilize linear chain conditional random fields (CRFs)[7] as our recognition model based on semantic features and dictionaries (NCBI taxonomy dictionary and UniProt taxonomy identifier). Also, we expanded our dictionary in some ways to deal with the special cases that influence our system to recognize the taxonomy entities. Second, taxonomy normalization. In this stage, we not only use the dictionary to identify the entities but also find the target species of each article to enhance the ability to normalize the species mentions.

A. Named Entity Recognition

To train a taxonomy named entity recognition model, we leverage the CRF++ toolkit (https://taku910.github.io/crfpp/).

In this model, we utilized BIEO states (B: begin, I: inside, E: end, O: outside) to tag each word and using a template of CRF which assists our model to recognize a taxonomy named entity. For training the CRF model, we need to get all the features of each token in the preprocessing stage.

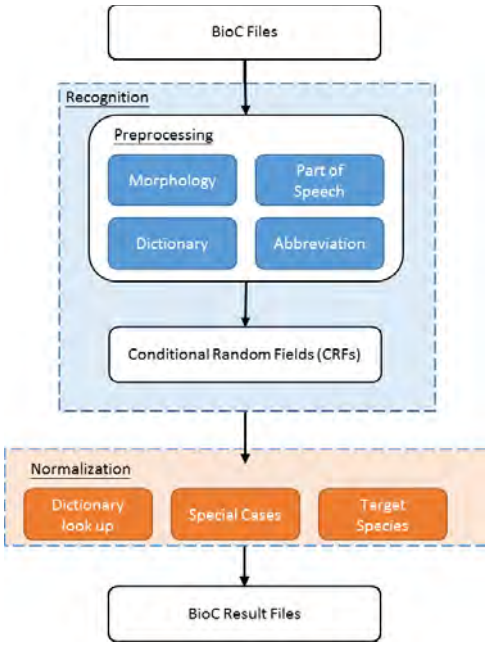


Fig. 1. *The architecture of our system, which can be divided into two stages, recognition stage and normalization stage.*

First, we divide tokens with not only spaces and punctuations but also letters and digits. After getting all the tokens, we transform them into lowercase, and get several features of them. In AuDis[1], the author uses features like morphology and part of speech, which are really useful to recognize the meaning of each word, so in our model, we utilized these as some of our important features. The significant features are describe as below:

- Morphology: this feature includes the original tokens, stemmed tokens which are extracted by the Snowball library, and prefixes/suffixes of the tokens whose lengths range from 1 to 5.
- Part of Speech: we use Stanford POS tagger to extract the part of speech of the token.
- Dictionary: we utilize the dictionary from NCBI taxonomy and UniProt Taxonomic identifier, to get the token's rank, like species, genus etc.
- Abbreviation: if the curated entity is an abbreviation, we label it as abbreviation, otherwise, we label it as normal.

For the dictionary feature, there are two dictionaries that we use. In the dictionaries, there are several columns and information that can be used in our features, such as taxonomy ids, taxonomy names, other names, rank of each mention, scientific name and common names. In particular, to strengthen our dictionary lookup feature, we have two possible ways; first, if the token is one of the taxonomy named entities but can't be found in these two dictionaries, we add this token into our dictionary. Second, we add to our dictionary if the token is an abbreviation of a taxonomy named entity or some common abbreviation.

In addition to the methods above, sometimes the taxonomy named entities will appear in a word which should have a space between it but missing. We store these tokens as special cases and using these at the normalization stage to avoid the situation where the correct entity is found in the text but can't be found in the dictionary.

B. Conditional Random Fields module

CRFs have been applied in many entity extraction studies in biomedical literature. And CRFs are a type of discriminative undirected probabilistic model for computing conditional probability distributions. Lafferty, McCallum and Pereira define the conditional probability distribution p(YX) of a random variable Y given the input X as follows:

$$p(YX) = \frac{1}{Z(X)} exp\big(f(Y,X)\big) = \frac{\exp(f(Y',X))}{\sum_{Y'} \exp(f(Y',X))} \quad (1)$$

where Y = $\{y_1, y_2, \ldots, y_n\}$ is a label sequence from an observation sequence X = $\{x_1, x_2, \ldots, x_n\}$ which means a token sequence. Z(X) is the normalization term. To learn the feature weights in a CRF, we can use gradient ascent because it is memory efficient. The weighted feature function in equation 2 for deciding the label at position *i* is a function of the label at position *i*-1, and the entire observation sequence X, made up of all x_i, which are vectors of features.

$$f(Y,X) = \sum_{j=1}^{n} \sum_{i=1}^{m} w_i f_i(y_i y_{i-1}, X) \quad (2)$$

C. Normalization

After passing the data through our CRF model, we obtain the label of each token. With these labels, we can extract the taxonomy entities. At the taxonomy normalization stage, we use NCBI Taxonomy dictionary to identify each taxonomy named entity. If the taxonomy named entity cannot be found in the dictionary, the entity might be one of the special cases or it might be a kind of species' embryo, larva etc.

If it is the latter, it is a bioconcept ambiguity problem. To address this problem, we developed an approach for looking up the articles' target species. At first, we break the original full text article into tokens, and get the features of each token as in the preprocessing stage; next, we utilize our CRF model to extract the taxonomy entities in the article; finally, we choose the taxonomy entity which appear the most frequently in the full text article as the target species for this article.

TABLE I. CAPTIONS STATISTICS

Task Dataset	Captions
Training data	13696
Caption contains Taxonomy	4206

TABLE II. RESULTS WITH AND WITHOUT SPECIAL CASES

RUN	SC-			SC+		
	P	**R**	**F**	**P**	**R**	**F**
1	0.571	0.373	0.451	0.548	0.403	0.464
2	0.486	0.315	0.382	0.52	0.359	0.424
3	0.594	0.273	0.374	0.625	0.315	0.419
4	0.799	0.481	0.6	0.81	0.533	0.642
5	0.778	0.485	0.598	0.793	0.548	0.648
Avg.	0.645	0.385	0.481	0.659	0.432	0.519

With the target species of all the articles, we can identify the entities that are extensions of the target species. As a result, we can address the problem of taxonomy ambiguity, and this approach improves our system for taxonomy normalization.

Some species' names can cause ambiguity (i.e. mouse, mice), because the species like "mouse" can be identified as taxonomy ID:10088 as genus and taxonomy ID:10090 as species. To address this problem, we normalize the mentions like mouse, mice, rat which IDs are 10088 into 10090 to make it consistent, which means that we identify IDs for species rank, not genus rank.

IV. EXPERIMENT AND RESULTS

To make the data easier to train and raise the accuracy of recognition, we transform all the training data from BioC format into Pubtator format, and extract the annotated captions that contains taxonomy entities. The statistics of the captions are shown in Table I.

In our evaluation, we use 5-fold cross-validation to evaluate our result on training data that only contains annotated captions with taxonomy entities. The results are shown in Table II. The SC- means normalization without checking if the entity is one of the special cases, and the SC+ is using the special cases to identify the unknown taxonomy named entities in the normalization stage. This approach raises the F-measure about 0.03%.

Due to the better result of using special cases, we go on to analyze the effect of identifying the unknown entities using original articles' target species. As shown in Table III, the F-measure raises about 0.12%.

Finally, we get our test result from BioID scorer, Table IV shows our score in each corpus that contains scores with and without normalization. The score without normalization counts extracted taxonomy named entities, and the score with normalization counts the correct taxonomy IDs.

V. DISCUSSION AND CONCLUSION

From our test results, we found out that our performance on taxonomy named entities recognition is effective, but we have more room to improve in our normalization stage. We now discuss some methods that might improve the performance. First, the way we find the target species in the original articles, we only use our CRF model to detect the mentions in the article. A better way is to not only use our model but also use some

TABLE III. PERFORMANCE WITH TARGET SPECIES

RUN	SC+			SC+_Target		
	P	R	F	P	R	F
Avg.	0.659	0.432	0.519	0.703	0.579	0.635

other taxonomy identifier to enhance the accuracy of getting the right target species. Besides, we only use two dictionaries (NCBI Taxonomy dictionary and UniPort Taxonomy identifier) and only use NCBI Taxonomy dictionary for the normalization stage. As the results show, we need more high quality dictionaries to improve our normalization stage. Lastly, if an entity is correctly mentioned once, if the same word appears again, the word should be marked same as the entity above, as in AuDis's[1] post-processing. We believe that integrating this step will significantly raise our performance.

Overall, we found several features which can assist our system in developing enhanced CRF model for taxonomy named entities recognition. With these features, our system sees increased performance in the recognition stage. In the normalization stage, we also use several methods to correctly identify the mentions' IDs, such as finding the target species and the special cases which significantly improved the F-measure. In our future work, we will focus on enhancing the performance of the normalization stage with the methods mentioned above, and by considering some other machine learning approaches.

TABLE IV. TEST RESULTS FROM BIOID SCORER

	precision	*F-measure*	*norm_ precision*	*norm_ F-measure*
Any_Strict	0.813	0.701	0.609	0.555
Any_Overlap	0.853	0.735	0.609	0.555
Normalized_Strict	0.812	0.701	0.609	0.555
Normalized_Overlap	0.852	0.735	0.609	0.555

REFERENCES

1. Lee H.C. et al. An Enhanced CRF-Based System for Disease Name Entity Recognition and Normalization on BioCreative V DNER Task. Proc BioCreative Workshop. Sevilla, Spain, pp. 226–233, 2016.

2. Naderi N, Kappler T, Baker CJO, Witte R, "OrganismTagger: detection, normalization and grounding of organism entities in biomedical documents," Bioinformatics, vol. 27, pp.2721-2729, 2011.

3. R. Leaman et al. "TaggerOne: joint named entity recognition and normalization with semi-Markov models," Bioinformatics, vol. 32 pp. 2839–2846, 2016.

4. C.-H. Wei, H.-Y. Kao, and Z. Lu, "SR4GN: a species recognition software tool for gene normalization," Plos one, vol. 7, p. e38460, 2012.

5. M. Gerner, G. Nenadic, and C. M. Bergman, "LINNAEUS: a species name identification system for biomedical literature," BMC bioinformatics, vol. 11, p. 85, 2010.

6. J. Lafferty, A. McCallum, and F. C. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," 2001.

7. Z. Lu, "PubMed and beyond: a survey of web tools for searching biomedical literature, Database, vol. 2011, p. baq036, 2011.

8. Y.-Y. Hsu and H.-Y. Kao, "Curatable Named-entity Recognition using Semantic Relations," IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 12, pp. 785-792, 2014.

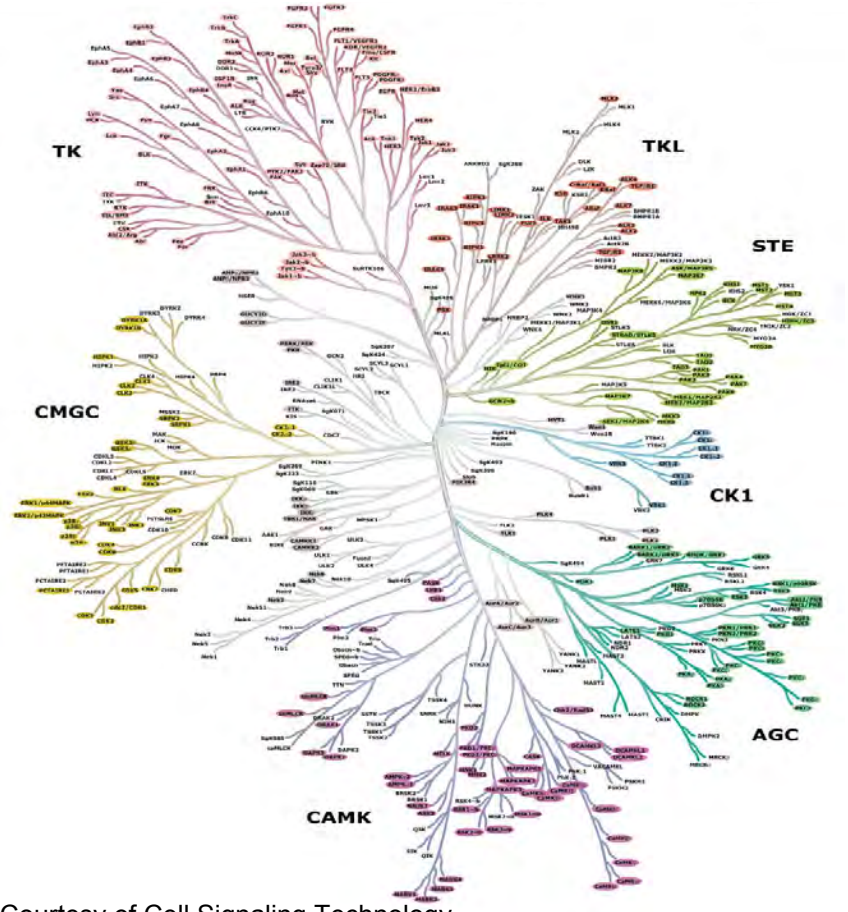
9. Doğan R.I. et al. NCBI disease corpus: A resource for disease name recognition and concept normalization. J. Biomed. Inf ., 47, 1‑10, 2014.

10. C.-H. Wei, H.-Y. Kao, and Z. Lu, "GNormPlus: An Integrative Approach for Tagging Genes, Gene Families, and Protein Domains," BioMed Research International, vol. 2015, 2015.

11. R. Leaman, C.-H. Wei, and Z. Lu, "tmChem: a high performance approach for chemical named entity recognition and normalization," Journal of cheminformatics, vol. 7, 2015.

12. J. G. Caporaso, W. A. Baumgartner, D. A. Randolph, K. B. Cohen, and L. Hunter, "MutationFinder: a high-performance system for extracting point mutation mentions from text," Bioinformatics, vol. 23, pp. 1862-1865, 2007.

Track 2
Text-Mining Services for Kinome Curation



Courtesy of Cell Signaling Technology

Overview of BioCreative VI Kinome Track

Text-mining services for Kinome Curation

Julien Gobeill[1,2], Pascale Gaudet[1], Patrick Ruch[1,2]
[1] SIB Swiss Institute of Bioinformatics, Geneva, Switzerland
[2] HES-SO / HEG Geneva, Information Sciences, Geneva, Switzerland

***Abstract*— The BioCreative VI Kinome Track proposed a competition to assess the effectiveness of text mining to perform literature triage, thanks to an unpublished curated dataset from the SIB Swiss Institute of Bioinformatics. This dataset contains comprehensive annotations for 300 human proteins kinases. For a given protein and a given curation axis (disease, or biological processes), participants' systems have to search relevant articles in a collection of 5.2M MEDLINE citations for the subtask 1, or 270,000 fulltext articles for the subfulltask 2. The list of explored strategies comprises Named Entity Recognition and machine learning frameworks. In particular, participants managed to derive a set of negative instances, as the databases typically do not store articles that were judged as irrelevant by curators. Final results shows significant improvements compared to the baseline established in a previous study, and compared to a basic PubMed search.**

Literature triage; protein curation; text mining

I. Introduction and motivation

Curators play a key role to define the content and ensure the quality of the biomedical databases and to spotlight the major findings (1,2). Their mission consists of continuously collecting, verifying and annotating the literature, in order to fill reference databases. Most curation methods are based on manual approaches, which produce the most accurate knowledge, but are time-consuming (3). With the exponential growth of biomedical literature (4), biocurators need help from the text mining community in order to remain up-to-date. In particular, (5) estimates that about 7% of the curation time is assigned to the rejection of papers. (6) assumes that 15% of curators' time is spent on triage. Assisting curators in retrieving, filtering and/or prioritizing the literature can save productivity.

The **CALIPHO group** develops the neXtProt database (7-8), a flagship resource of the SIB Swiss Institute of Bioinformatics that integrates information on human proteins. The data in the neXtProt database comes from both integration of external resources and annotation within the group using an internal annotation tool, the BioEditor. In a project funded by Merck-Serono from 2011-2013, the CALIPHO group has annotated 300 human protein kinases from over 13,600 research articles, producing a data corpus of over 30,000 different annotations describing the function, substrates of the kinases, and diseases in which they have been implicated. This large data corpus is still unpublished (to be released in the upcoming months), providing a unique opportunity to use curated data to create a text mining task.

Literature triage is an Information Retrieval task; it aims at retrieving/filtering articles that are supposed to be relevant for curation. This is a basic task performed by virtually all curated molecular biology databases to initiate a curation workflow. The BioCreative VI Kinome Track proposed a competition in literature triage, thanks to the neXtProt data. Text mining groups were invited to develop and test approaches aiming at assisting database curators in the selection and ranking of relevant articles for the curation of human protein kinases. Two aspects were investigated in two subtask: abstracts triage, and fulltexts triage. All abstracts annotated in the neXtProt data are available via MEDLINE. The availability of fulltexts is more problematic, as only a tiny fraction (approximately 10%) is open access in services such as Europe PMC (Europe PMC consortium).

II. Tasks & Data

A. The Kinome Track dataset

The BioCreative VI Kinome Track dataset contains comprehensive annotations about kinase substrates. It covers a significant fraction of the Human Kinome: 300 proteins out of approximately 500 human kinases. The dataset contains more than 30,000 annotations, all supported by an article (a PubMed identifier). The Kinome Track focuses on two different curation axes: diseases, and biological processes. This dataset is ready to be integrated in the neXtProt database by 2017, yet still hidden from public and participants during the competition period.

The dataset represents a total of 4,581 different articles annotated for diseases annotations, and 5,357 for biological processes ones. There is a slight overlap between both axes: only 6% of the curated articles contain an annotation for both a disease and a biological process. In total, 9,367 different articles, published in 862 different journals, are present in the dataset. Table 1 shows the 10 most represented journals in the dataset.

TABLE I. Top 10 Journals in dataset

Journal	# articles in dataset	Cumulative percentage
J Biol Chem	744	7.9%
Proc Natl Acad Sci U S A	314	11.3%
Cancer Res	301	14.5%
Blood	288	17.6%

Journal	# articles in dataset	Cumulative percentage
Mol Cell Biol	253	20.3%
Oncogene	228	22.7%
PLoS One	219	25.1%
J Immunol	208	27.3%
Nature	156	28.9%
Clin Cancer Res	156	30.6%

Fig. 1. The top 10 journals in the dataset, ranked by the number of articles annotated in the dataset. The cumulative percentage is computed for the whole collection (e.g. *the top 10 journals represent 30.6% of all the annotated articles in the dataset*).

B. *The Kinome Track benchmark*

In the Cranfield paradigm (9) for evaluation of Information Retrieval systems, benchmarks are composed of three parts: a collection of documents, a set of queries, and relevance judgements. In the Kinome Track, a query was a human kinase and an axis (biological processes, or diseases). Participants' systems had to search in the collection, and to propose a ranked list of articles that are relevant for the curation of this kinase and this axis. Systems were evaluated on their ability to propose the articles that were chosen by neXtProt curators.

a) Design of the collection: For a fair comparison, systems obviously had to search for relevant articles in a common collection. This collection had to satisfy two conditions: being small enough to be efficiently processed by all teams for the competition, but large enough to make the task realistic. For designing such a collection, we applied a journal-centric strategy. As mentioned previously, annotated articles in the dataset were published in 862 different journals (out of approximately 5,500 in MEDLINE). We chose to include all articles published in these 862 journals. We also decided to limit the collection to papers published before 2014. Thus, the final collection contained a total of 5.3M PMIDs. Unfortunately, for the fulltexts collection, only a small fraction of the PMIDs corresponded to an open-access article.

b) Queries: They were pairs made of: one of the 300 kinases (e.g. "Activin receptor type-1B" - P36896), and a curation axis (biological processes, or diseases). For each kinase, neXtProt collected synonyms were provided to the participants.

c) Relevance Judgements: For a given kinase and a given axis, all annotated articles in the dataset are considered as relevant. The rejections of articles after screening by a curator are not stored in the database. Yet, the neXtProt curation is assumed to be comprehensive. This assumption means that all potential articles were screened, and that articles that were not annotated are considered as non-relevant. Articles that could have never been screened by a curator are equally distributed among all participants runs, and thus comparisons between systems are still valid.

C. *The Kinome Track subtasks*

Figure 2 presents an overview of the triage process.

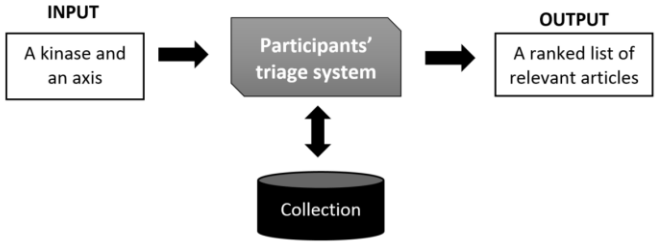


Fig. 2. Overview of literature triage for the Kinome Track. (*the collection differs depending on the subtask: abstracts, or fulltexts*).

The 300 proteins present in the dataset were randomly distributed in three different subsets: one third for the tuning set, one third for the subtask 1 test set, and one third for the subtask 2 test set. The tuning set contained 100 kinases, along with the PMIDs of the annotated articles for each axis (relevance judgements). The tuning set was made available in April 2017, thus participants were free to use it for analyses, and for tuning their system. Both test sets were delivered in May 2017. Obviously, test sets only contained the queries, while relevance judgements were kept for the official evaluation.

a) Subtask 1 – abstracts triage: This subtask focuses on abstracts triage. The collection (5.3M of articles) was given in the form of MEDLINE citations. Thus, systems had to perform triage only based on abstracts, and metadata (such as journal, publication year, publication type, etc). In the test set, for each kinase, this collection contained on average 16 relevant articles for the biological process axis, and 18 for the diseases axis. The collection was provided in BioC format.

b) Subtask 2 – fulltexts triage: This subtask focuses on fulltexts triage. The collection was given in the form of PubMed Central (PMC) fulltexts. Thus, systems had to perform triage based on fulltext contents. As only a fraction of PMC is open-access, the collection for the subtask 2 only contained 260,000 articles. Thus, in the test set, for each kinase, this collection only contained on average 2.6 relevant articles for the biological process axis, and 3.6 for the diseases axis. The collection was provided in XML format.

D. *Metrics used for evaluation*

TREC formats and metrics were used for evaluation (10) as follows:

- P10 or Precision at rank 10: among the top 10 articles returned by the system, how many are relevant. If the system returned 10 documents and only 4 are relevant, then P10 is 0.4. Idem for P30, P100 at ranks 30 and 100.
- R30 or Recall at rank 30: among all the relevant articles in the collection for a given query, how many are retrieved in the top 30 articles returned by the system. If for a given query there are 20 relevant documents in the

collection, and the system returns 10 of them in the top 30 documents, then R30 is 0.5. Idem for R100 at rank 100.

- P at R0: this is the maximum precision observed (for any rank value).
- Mean Average Precision (MAP): this is the average of all Precision at rank k, for ranks where a relevant article is retrieved (for no retrieved articles, 0 is counted).
- R-Prec: this is the Precision observed at rank r, where r is the number of relevant articles for a given query. If for a given query there are 20 relevant articles in the collection, R-Prec is Precision at rank 20.

III. RESULTS

More than twenty teams registered to the Kinome Track, but two finally submitted runs. In this section, we broadly describe strategies investigated by these participants. Then, we give results for both subtasks. All metrics were computed with the trec_eval reference program (http://trec.nist.gov/trec_eval/).

A. Participants strategies

Both participating teams exploited tuning data in order to train a machine learning system. They applied Named Entity Recognition system – such as the PubTator system (11) – in order to identify biological processes and diseases in the relevant articles, and thus obtained a set of positive pairs of kinase-concept. One team exploited these positive examples and the rest of the articles collection in order to bootstrap some pseudo negative samples. Then, both teams trained a machine learning system for triage based on several features, such as numbers and position of genes and axis.

B. Subtask 1 – abstracts triage

For the subtask 1, each team could submit up to ten runs. There were twenty runs for the disease axis, and nineteen for the biological process axis. Figures 3 and 4 present results for each run.

Team	Run name	MAP	R-Prec	P at R0	P10	P30	P100	R30	R100
383	DIS_team383run3	0.109	0.147	0.458	0.152	0.098	0.052	0.222	0.327
383	DIS_team383run8	0.109	0.145	0.453	0.148	0.097	0.052	0.223	0.327
383	DIS_team383run4	0.108	0.142	0.455	0.151	0.098	0.052	0.225	0.326
383	DIS_team383run5	0.088	0.125	0.351	0.119	0.081	0.044	0.203	0.304
383	DIS_team383run6	0.088	0.125	0.351	0.117	0.081	0.044	0.201	0.304
383	DIS_team383run1	0.081	0.098	0.37	0.103	0.075	0.042	0.184	0.286
383	DIS_team383run7	0.079	0.099	0.338	0.103	0.075	0.042	0.182	0.288
383	DIS_team383run2	0.073	0.084	0.338	0.094	0.064	0.038	0.166	0.269
383	DIS_team383run9	0.062	0.079	0.224	0.075	0.054	0.036	0.15	0.265
383	DIS_team383run10	0.06	0.079	0.227	0.065	0.057	0.034	0.154	0.259
392	run10_Abst_HPO_DIS_L-BOW	0.098	0.134	0.37	0.138	0.087	0.047	0.196	0.304
392	run6_Abst_NCIT_DIS_L-BOW	0.096	0.134	0.384	0.14	0.087	0.047	0.199	0.302
392	run7_Abst_NCIT_DIS_Rel	0.08	0.107	0.319	0.118	0.074	0.042	0.173	0.269
392	run26_Abst_NCIT_DIS_Ag	0.069	0.094	0.305	0.104	0.068	0.04	0.159	0.259
392	run28_Abst_HPO_DIS_Ag	0.053	0.073	0.235	0.079	0.054	0.033	0.128	0.23
392	run9_Abst_NCIT_DIS_S-ENG	0.051	0.077	0.176	0.069	0.059	0.034	0.146	0.237
392	run19_Abst_HPO_DIS_Rel	0.049	0.075	0.215	0.07	0.052	0.032	0.122	0.227
392	run11_Abst_HPO_L-ENGA	0.048	0.056	0.199	0.053	0.043	0.028	0.109	0.206
392	run8_Abst_NCIT_DIS_L-ENGA	0.044	0.059	0.153	0.06	0.044	0.028	0.117	0.209
392	run12_Abst_HPO_S-ENG	0.04	0.054	0.164	0.054	0.041	0.024	0.11	0.186

Fig. 3. Results for the abstracts triage subtask, disease axis.

Team	Run name	MAP	R-Prec	P at R0	P10	P30	P100	R30	R100
383	BP_team383run3	0.195	0.182	0.45	0.176	0.121	0.065	0.399	0.563
383	BP_team383run8	0.192	0.184	0.43	0.171	0.122	0.064	0.397	0.563
383	BP_team383run4	0.191	0.178	0.437	0.171	0.122	0.064	0.396	0.561
383	BP_team383run5	0.172	0.168	0.379	0.143	0.107	0.057	0.361	0.526
383	BP_team383run6	0.17	0.169	0.378	0.14	0.107	0.057	0.361	0.524
383	BP_team383run1	0.159	0.15	0.379	0.138	0.105	0.057	0.362	0.535
383	BP_team383run2	0.155	0.141	0.373	0.137	0.104	0.056	0.346	0.529
383	BP_team383run9	0.127	0.109	0.251	0.086	0.074	0.045	0.292	0.468
383	BP_team383run7	0.119	0.109	0.242	0.101	0.077	0.046	0.285	0.457
383	BP_team383run10	0.109	0.078	0.219	0.075	0.064	0.044	0.266	0.455
392	run1_Abst_GO_BP_L-BOWA	0.201	0.21	0.466	0.184	0.111	0.056	0.36	0.492
392	run31_Abst_GO_BP_Rel	0.197	0.189	0.433	0.17	0.117	0.06	0.379	0.529
392	run2_Abst_GO_BP_L-BOW	0.187	0.185	0.438	0.172	0.108	0.057	0.351	0.503
392	run30_Abst_GO_BP_Ag	0.181	0.172	0.383	0.158	0.112	0.058	0.372	0.519
392	run3_Abst_GO_BP_L-ENGA	0.18	0.174	0.399	0.152	0.098	0.052	0.347	0.494
392	run33_Abst_GO_BP_K	0.175	0.166	0.364	0.149	0.109	0.058	0.368	0.52
392	run32_Abst_GO_BP_Prx	0.154	0.156	0.355	0.145	0.095	0.051	0.335	0.482
392	run4_Abst_GO_BP_S-ENG	0.145	0.141	0.311	0.124	0.081	0.049	0.315	0.477
392	run34_Abst_GO_BP_A	0.13	0.115	0.293	0.092	0.062	0.035	0.272	0.39

Fig. 4. Results for the abstracts triage subtask, biological process axis.

In (12), the Text Mining group at the SIB describes the development of neXtA5, a curation service and interface, powered by different ontologies, and developed for the CALIPHO group. This system aims at assisting biocurators by prioritizing articles for the curation of a given protein and a given axis. The system was evaluated on the same dataset than in the Kinome Track. The study also compared the neXtA5 ranking with PubMed ranking (both Boolean and relevance-based). For the disease axis, the reported values P at R0 were between 0.12 and 0.13 for PubMed rankings, while they reached up to 0.41 for neXtA5. The best reported MAP was 0.04. For the biological process axis, the reported values P at R0 were 0.14 for both PubMed, and 0.45 for neXtA5. The best reported MAP was 0.11. These values can be considered as the baselines for the interpretation of results in the Kinome Track.

In terms of P at Ro values, for the disease axis, three runs (submitted by the same team) makes better than the baseline for P at R0 (+10%). For these runs, on the top 100 returned articles, 22% are relevant (P100), and they represent 33% of all available relevant articles (R100). For the biological process axis, best results observed are close to the baseline for P at R0, while the second team made slightly better. Even if we don't know yet the details of the implemented strategies, it is interesting to note that the best runs submitted by team 383 for both axes share the same names. In (Mottin 2016), separate strategies were implemented for each axis. The main advances are observed regarding the recall with a significant improvements at MAP values (+175% for disease, +80% for biological processes).

C. Subtask 2 –triage of fulltext articles

For the subtask 2, each team could submit up to ten runs. There were ten runs for the disease axis, and eight for the biological process axis. Figures 5 and 6 present results for each run.

Team	Run name	MAP	R-Prec	P at R0	P10	P30	P100	R30	R100
392	run27_FT_NCIT_DIS_Ag	0.118	0.111	0.278	0.065	0.028	0.011	0.233	0.293
392	run23_FT_NCIT_DIS_Rel	0.112	0.105	0.269	0.064	0.03	0.011	0.24	0.296
392	run25_FT_HPO_DIS_Rel	0.1	0.103	0.227	0.052	0.026	0.011	0.207	0.295
392	run29_FT_HPO_DIS_Ag	0.097	0.083	0.219	0.058	0.029	0.011	0.229	0.271
392	run24_FT_HPO_DIS_L-BOW	0.091	0.078	0.193	0.043	0.025	0.011	0.22	0.302
392	run22_FT_NCIT_DIS_L-BOW	0.088	0.076	0.193	0.038	0.025	0.011	0.219	0.306
392	run17_FT_HPO_DIS_L-ENGA	0.083	0.083	0.172	0.046	0.023	0.01	0.19	0.274
392	run16_FT_NCIT_DIS_S-ENG	0.041	0.028	0.076	0.018	0.013	0.007	0.124	0.212
392	run18_FT_HPO_DIS_S-ENG	0.041	0.024	0.09	0.025	0.015	0.007	0.141	0.217
392	run15_FT_NCIT_DIS_L-ENGA	0.029	0.017	0.056	0.012	0.013	0.007	0.127	0.194

Fig. 5. Results for the fulltexts triage subtask, disease axis.

Team	Run name	MAP	R-Prec	P at R0	P10	P30	P100	R30	R100
392	run35_FT_GO_BP_Ag	0.293	0.203	0.349	0.079	0.037	0.014	0.61	0.714
392	run38_FT_GO_BP_K	0.278	0.194	0.327	0.081	0.036	0.014	0.616	0.728
392	run37_FT_GO_BP_Prx	0.271	0.182	0.314	0.073	0.037	0.014	0.624	0.712
392	run40_FT_GO_BP_Prx50	0.267	0.165	0.313	0.078	0.036	0.014	0.608	0.712
392	run36_FT_GO_BP_Rel	0.264	0.169	0.312	0.075	0.037	0.014	0.631	0.724
392	run13_FT_GO_BP_L-ENGA	0.042	0.004	0.053	0.017	0.012	0.007	0.232	0.461
392	run39_FT_GO_BP_A	0.03	0.002	0.038	0.008	0.008	0.007	0.188	0.423
392	run14_FT_GO_BP_S-ENG	0.021	0	0.026	0.005	0.006	0.006	0.17	0.365

Fig. 6. Results for the fulltexts triage subtask, biological process axis.

Comparisons between abstracts and full text articles results must be made with great care. Indeed, the sizes of the collections made the direct comparison difficult (5.2M in subtask 1 versus 260,000 in subtask 2). The number of relevant articles was 16-18 per query in subtask 1 versus 2-3 per query in subtask 2. In this perspective, the best reported MAP values are higher with full text than abstracts, but the likeliness to miss some relevant articles in the subtask 1 is higher – and so to be penalized – than in subtask 2. A comparison remains possible using P at R0 since it focuses on the precision of the first relevant retrieved article. Best reported P at R0 values are higher with abstracts than fulltexts.

D. Subtask 3 – snippets extraction

A third subtask was initially considered: snippet extraction. In this subtask, the participants' system should extract from the fulltext a snippet of maximum 500 characters, which contains enough information to be "annotatable". Curators should judge snippets according to one of the three following values: 1 = Good (the snippet is sufficient for making an annotation without reading the paper); 0.5 = quite good (the curator thinks that there is a potential annotation, but needs to read the paper because the snippet is not sufficient for making the entire annotation); 0 = Irrelevant (nothing in the snippet indicates that an annotation is possible). Unfortunately, this task was cancelled due to no submitted runs.

IV. Discussion and Conclusion

Participants' papers and presentations will describe the respective methods. In particular, the impact of the bootstrapping strategy proposed by one of the competitor to acquire negative instances seems especially effective. Altogether, the precision at P0 improves slightly over (12) on the same tasks but with a different dataset, while the MAP is more significantly improved. Moreover, the triage of articles to support the curation of biological processes shows better results than for curation of diseases (+82% for best MAP values).

The low ratio of participation for registered teams is difficult to explain. The huge – yet realistic – size of the collection (5.2M of articles) could have been a serious obstacle. Moreover, data provided to participants did not contain any annotated concepts. Yet, both teams developed supervised strategies based on the potential concepts present in the articles. Thus, they had no gold standard for building their knowledge base, and had to use NER systems in order to generate a silver standard.

Further, data-driven approaches do need positive and negative examples for training. Unfortunately, curated databases usually do not store negative examples, such as articles, which are screened by the curator but not selected for curation. Hopefully, such a data stewardship "gap" should improve as FAIR principles become common practice.

Acknowledgment

This work was supported by the Swiss National Fund for Scientific Research, SNF Grant: [neXtPresso project, SNF #153437].

References

1. Mottin, L., Pasche, E., Gobeill, J., Rech de Laval, V., Gleizes, A., Michel, P. A., ... & Ruch, P. (2017). Triage by ranking to support the curation of protein interactions. Database, 2017(1).

2. Burge,S., Attwood,T.K., Bateman,A. et al. (2012) Biocurators and biocuration: surveying the 21st century challenges. Database (Oxford), 2012, bar059. doi: 10.1093/database/bar059.

3. Hirschman,L., Burns,G.A.P.C., Krallinger,M. et al. (2012) Text mining for the biocuration workflow. Database, 2012, bas020.

4. Lu,Z. (2011) PubMed and beyond: a survey of web tools for searching biomedical literature. Database, 2011, baq036.

5. Wiegers,T.C., Davis,A.P., Cohen,K.B. et al. (2009) Text mining and manual curation of chemical-gene-disease networks for the Comparative Toxicogenomics Database (CTD). BMC Bioinformatics, 10, 326.

6. Gaudet, P.; Michel, P. A.; Zahn-Zabal, M.; Cusin, I.; Duek, P. D.; Evalet, O.; Gateau, A.; Gleizes, A.; Pereira, M.; Teixeira, D.; Zhang, Y.; Lane, L.; Bairoch, A. The neXtProt knowledgebase on human proteins: current status. Nucleic Acids Res. 2015, 43, D764-70, doi:10.1093/nar/gku1178.

7.. Gaudet, P.; Michel, P.-A.; Zahn-Zabal, M.; Britan, A.; Cusin, I.; Domagalski, M.; Duek, P. D.; Gateau, A.; Gleizes, A.; Hinard, V.; Rech de Laval, V.; Lin, J.; Nikitin, F.; Schaeffer, M.; Teixeira, D.; Lane, L.; Bairoch, A. The neXtProt knowledgebase on human proteins: 2017 update. Nucleic Acids Res. 2017, 45, D177–D182, doi:10.1093/nar/gkw1062.

8. The Europe PMC Consortium (2014) Europe PMC: a full-text literature database for the life sciences and platform for innovation. Nucleic Acids Res., 43, D1042–D1048.

9.Cleverdon, C. W. (1960, December). The ASLIB Cranfield research project on the comparative efficiency of indexing systems. In Aslib Proceedings (Vol. 12, No. 12, pp. 421-431). MCB UP Ltd.

10. Manning, C., Raghavan, P., & Schütze, H. (2008). Evaluation in information retrieval. In Introduction to Information Retrieval (pp. 139-161). Cambridge: Cambridge University Press. doi:10.1017/CBO9780511809071.009

11. Wei, C. H., Kao, H. Y., & Lu, Z. (2013). PubTator: a web-based text mining tool for assisting biocuration. Nucleic acids research, 41(W1), W518-W522.

12. Mottin, L., Gobeill, J., Pasche, E., Michel, P. A., Cusin, I., Gaudet, P., & Ruch, P. (2016). neXtA 5: accelerating annotation of articles via

automated approaches in neXtProt. Database, 2016, baw098.Mottin et al.

Assisting Document Triage for Human Kinome Curation via Machine Learning

Yi-Yu Hsu, Chih-Hsuan Wei and Zhiyong Lu*
Affiliation: National Center for Biotechnology Information, Bethesda, MD 20894, USA
*Correspondence: zhiyong.lu@nih.gov

***Abstract*—Manual curation efforts are often tedious and have largely been limited by the need to process and integrate large numbers of biomedical literature in finite amounts of time. Document triage with automatic methods can compensate for the weakness of manual curation efforts and help provide more efficient and effective curation workflows. Machine-assisted document triage involves automatic identification of entities and relation extraction with natural language processing techniques. Here, we developed a system that can automatically predict which articles have a higher relevance for curation. In the triage task for Human Kinome Curation in BioCreative (BC) VI, we applied several machine learning methods for exploring articles with curatable knowledge, in particular the bio-concept relations among kinases, diseases, and biological processes. We used rich co-occurrence and linguistic features to assess the influence of human kinome articles from the neXtProt database. We expect this method can provide neXtProt biocurators with ranking lists for specific queries, thus facilitating the process of reviewing relevant information in the literature.**

Keywords—document triage, machine learning, relation extraction

I. Introduction

Assisting biocurators in the retrieval of relevant articles and passages for the curation of protein kinases proves to be an ongoing challenge. The BC VI Human Kinome Curation Track addresses this problem and is designed as an information retrieval task (literature triage) aimed at retrieving relevant articles for specific curation efforts (i.e. biological process (BP)/disease (DIS)). To help develop and evaluate approaches for this task, the neXtProt data (cite 25593349) is used and includes 300 proteins protein kinases. The articles in this database contain comprehensive manual annotations including gene ontologies, biological processes, and the National Cancer Institute (NCI) diseases.

This paper describes our submission to the human kinome curation track at the BC VI. This track includes three subtasks: (1) abstract triage, (2) full text triage, and (3) snippet selection. We participated in the first subtask, which is about retrieving curatable articles based on abstracts. As biocurators spend significant amounts of time surveying and reviewing articles using specific queries, a precise document triage classification could be helpful in reducing the workload of biocurators and allowing them to customize their own curation patterns (1, 2).

II. System Description

A. Data preprocessing

For the training set, the BC VI organizers provided two datasets, each including 1,615 and 1,844 pairs (<kinase, PMID>) with its associated axis, which can either be a biological process or disease. However, the datasets do not annotate which biological processes or diseases correspond to the annotated kinase. In this study, we combined the two sets (1,615 + 1,844 = 3,459) and generated triples <kinase, axis, PMID>. For instance, there is a relationship between "SGK1" and "myeloma" in Fig. 1., which would be noted as <SGK1, myeloma, 21478911> in the triple.

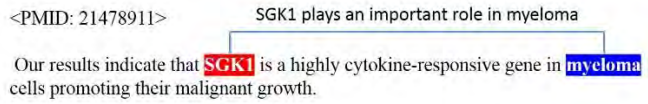


Fig. 1 An example of positives in the training set

First, we used our Named Entity Recognizer (NER) taggers (3, 4) to recognize all kinase, disease, and biological process mentions. We filtered out the articles without kinase mentions, and narrowed our results to 2,775 triples. In order to evaluate our method, we kept 225 triples as a development set, leaving 2,550 triples for training (Table 1).

We also selected articles with 100 target proteins from 5.3 million citations, therefore creating 894,312 triples. Although there are no negative training instances in the datasets provided, we generated pseudo-negatives by using the following process. First, we used a Support Vector Machine (SVM)-one class classifier to train on 2,550 triples and test on 894,312 triples, and then selected the lowest 2,500 scores as our negative training instances. Note that we now have a positive set (2,550 triples) and a negative set (2,500 triples). We then trained our models on different classifiers described in the methods section using both the 2,550 positive set and 2,500 negative set. After the models were built, we added the 225 triples to the 894,312 triples to evaluate the ranking scores of the development set. Fig. 2 shows the workflow of our proposed document classification system for human

kinome curation using machine learning.

Table 1. Statistics of the training set

	Official Training set	Utilized	
		Training set	Development set
# Triple	3,459	2,550	255
# PMID	3,018	2,282	221

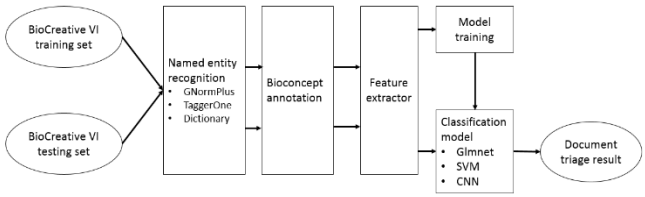


Fig. 2 The workflow of our human kinome curation system

B. *Methods*

Our submission to the abstract triage task utilizes several machine learning methods including lasso (least absolute shrinkage and selection operator) and Elastic-Net Regularized Generalized Linear Models (Glmnet) (5), SVM (6), and Convolutional Neural Network (CNN) (7). As mentioned, we applied our bio-concept taggers (TaggerOne (3) and GNormPlus (4)) to recognize all disease and kinase mentions, and then built a dictionary-based tagger to annotate biological process mentions. The performance of bio-concept recognition on the training set is shown in Table 2. Overall, the performance of the target kinase recognition is lower than the performance of the bio-concept recognition. When the recognition rate of DIS and BP mentions achieves over 98%, the recognition rate of the target kinase achieves less than 80% in recall. In our observation, 60% of missed target kinases are not mentioned in the abstract. The input data for the machine learning based models includes only the title, abstract, and bio-concept annotations of the taggers. Additionally, our methods did not distinguish between the data for disease and biological process mentions. The texts of the two types were trained together by using the same features.

Table 2. The NER results of target kinase, DIS, and BP

Mentions	Set	Method	Recall	Found	All
DIS	DIS_train.qrel	TaggerOne	0.9807	1375	1402
BP	BP_train.qrel	Dictionary	0.9975	1612	1616
Target Kinase	DIS_train.qrel	GNormPlus	0.7879	1304	1655
	BP_train.qrel		0.6302	1162	1844

- *Glmnet*

The features of large datasets would suffer from a curse of dimensionality, and they usually generate large sparse data matrices. To reduce the high dimensional features, Glmnet is a widely used algorithm for fitting various probability distributions in statistical computing and machine learning. When analyzing the high dimensional data, Glmnet uses the lasso or the elastic net to interpret and fetch important features with efficient computation. Therefore, Glmnet increased in stability and made predictions with a path of penalty parameters.

- *SVM*

A Support Vector Machine (SVM) is a robust machine learning algorithm for classification analysis. The SVM has been applied to many classification problems related to supervised learning with multidimensional data. After the SVM classifier is built, the model can correctly determine the hyperplane, which separates the data into different classes.

  - One-class classification: this model aims to find the support vectors of the one-class training set, and allows for outlier/novelty detection (8). The goal is to distinguish new data as either similar or different from the normal training set.
  - Binary classification: The original SVM is designed for determining the optimal separating hyperplane between the two groups. In practice, the SVM project samples on a higher dimensional space to approach the optimal hyperplane with less empirical classification errors (6).

- *CNN*

A Convolutional Neural Network (CNN) is derived from deep artificial neural networks that consist of receptive fields, local connectivity, and shared weights (7). The CNN has been well known for its excellent performance on image recognition. In this work, we trained a simple CNN with two layers of convolution on top of word vectors obtained from an unsupervised learning algorithm (9). We then designed and aligned the CNN with different parameters including an input layer, convolution layer, pooling layer, fully connected layer, and output layer.

- *Features*

The following features are applied to all the methods, as shown in Table 3. The features can be grouped into three categories: A) frequency features (feature 1-2): calculated the number of kinase and axis mentions in each abstract. B) location features (feature 3-7): the location of kinase and axis is detected. C) natural language processing (NLP) features (feature 8-11): kinase key words include a list of keyword groups, which is shown in Table 4. Each group includes manually generated key words of the genetic disease field. Furthermore, we applied tmVar (10, 11) to recognize variation mentions in the text as an additional variation key word group. The bag of words feature includes the lemma form of words around kinase, disease, and biological process mentions in the abstracts. Parsing tree path features use the dependency relation of dependency grammars to record the syntactic structure of kinase, disease, and biological process mentions (12). All features are transformed to document-term matrices.

Table 3. Statistical and linguistic features

Feature		Type
1	Number of target kinase	Numeric
2	Number of target axis	Numeric
3	Target kinase in 1st sentence	Boolean
4	Target axis in 1st sentence	Boolean
5	Target kinase in last sentence	Boolean
6	Target axis in last sentence	Boolean
7	The same sentence	Boolean
8	Kinase key words	String
9	Bag of words	String
10	Parsing tree path	String
11	Parsing tree path w/o ancestors	String

Table 4. The keyword groups

Group	Key words
Verb	involve, enhance, inhibit, regulate, increase, associate, phosphorylate
Patient	patient, men, women
genetic	detectable, survival, genetic, tumorigenesis, overexpression, mutation, translate, transcript, change, lymphangiogenic, neurotrophic
scale	mg, kg
period	day(s), during
examine	examine, experiment, screen, role, risk, significant
Variation	recognized by tmVar

EVALUATION

Before submitting official runs, we used the following evaluation metrics to assess each of our models:

- MAP (Mean Average Precision) is the mean of the precision scores for various queries.
- We also defined an estimated score ($Escore$) for measuring the ranking result of a triple (t) including a kinase, an axis, and a PMID. Note that γ is the rank of a triple after we combined the triples of the training and development set ($|D|$). We then summarized the score $\frac{\gamma}{|D|}$ of all 225 triples from the development set. For example, if we assume there are 10 PMIDs mentioned for a target kinase and the rank of one specific PMID is the top one among all 10 PMIDs, then $\frac{\gamma}{|D|}$ is 0.1. Therefore, the lower $Escore$ represents a better performance.

$$Escore = \sum_{|t|} \frac{\gamma}{|D|}$$

The following evaluation metrics are used by organizers for official results.

- P10 is the precision at rank 10: we calculated the number of documents that are relevant among the top ten documents returned by the system. If the system returns ten documents and only four documents are relevant, the P10 is 0.4. Also, P30 and P100 are the precision at rank 30 and 100 respectively.
- R30 is the recall at rank 30: we calculated the number of relevant documents retrieved in the top 30 documents returned by the system. We assume that within each query there are only 20 relevant documents within the collection. If the system returns ten of these relevant documents, then the R30 is 0.5. Also, R100 is the recall at rank 100.
- P at R0 is the maximum precision observed (for any rank value).
- R-Prec is the precision observed at rank r, where r is the number of relevant documents in the collection. If a given query contains twenty relevant documents, R-Prec is the precision at rank 20.

As shown in Table 5, we trained different models with features described in Table 3. For the Glmnet classifiers, both BP and DIS triples are included in the training set. An SVM (binary) is the model that we applied both positives and negatives as a binary classifier, while an SVM (one class) uses only the positives to train a one-class classifier. For the CNN classifiers, we constructed multiple hidden layers between the input and output layers, and modeled complex non-linear relationships. The evaluations of different methods on training sets (including disease and biological process sets) are reported in Table 6. In the last stage, we used the entire positives in the training set and features of better performance in our evaluation for method 9 and 10. In this case, we did not have the testing triples to evaluate both models. Overall, the performance of the Glmnet classifiers is superior compared to the other two classifiers. After reviewing and optimizing the parameters in the training set, we then used the following methods and features in Table 5 as our ten submitted runs.

Table 5. Combinations of different methods and features

Method	Features	Positives of Training set	Classifier
1	9	2500	Glmnet
2	1-9	2500	Glmnet
3	1-10	2500	Glmnet
4	1-9, 11	2500	Glmnet
5	1-10	2500	SVM (Binary)
6	1-9, 11	2500	SVM (Binary)
7	1-9, 11	2500	SVM (One class)
8	1-9, 11	2500	CNN
9	1-9, 11	2775	Glmnet
10	1-9, 11	2775	CNN

Table 6. Evaluation of training set

Method	$Escore$	MAP
1	58.09	0.0401
2	49.84	0.0535
3	46.46	0.0593
4	49.85	0.0598

5	52.88	0.0460
6	57.32	0.0470
7	82.20	0.0227
8	83.68	0.0204

Table 7 and 8 demonstrate the official results of the abstracts triage (provided by the task organizers). Both tables show that Glmnet classifiers have a better performance of MAP than that of SVM and CNN classifiers, which is consistent with our observation in the training & development phases. When considering the *Escore* metrics, Glmnet classifiers are also consistent with the best performance compared to all classifiers. As for CNN classifiers, the approach sketched here fails to compete with other classifiers because when running on a small dataset an over-fitting problem develops. CNN classifiers might be able to identify more relations using larger datasets.

Table 7. Official results of kinases/diseases for the ten submitted runs.

Method	MAP	R-Prec	P at R0	P10	P30	P100	R30	R100
3	0.109	0.147	0.458	0.152	0.098	0.052	0.222	0.327
9	0.109	0.145	0.453	0.148	0.097	0.052	0.223	0.327
4	0.108	0.142	0.455	0.151	0.098	0.052	0.225	0.326
5	0.088	0.125	0.351	0.119	0.081	0.044	0.203	0.304
6	0.088	0.125	0.351	0.117	0.081	0.044	0.201	0.304
1	0.081	0.098	0.370	0.103	0.075	0.042	0.184	0.286
7	0.079	0.099	0.338	0.103	0.075	0.042	0.182	0.288
2	0.073	0.084	0.338	0.094	0.064	0.038	0.166	0.269
8	0.062	0.079	0.224	0.075	0.054	0.036	0.150	0.265
10	0.060	0.079	0.227	0.065	0.057	0.034	0.154	0.259

Table 8. Official results of kinases/biological processes for the ten submitted runs.

Method	MAP	R-Prec	P at R0	P10	P30	P100	R30	R100
3	0.195	0.182	0.450	0.176	0.121	0.065	0.399	0.563
9	0.192	0.184	0.430	0.171	0.122	0.064	0.397	0.563
4	0.191	0.178	0.437	0.171	0.122	0.064	0.396	0.561
5	0.172	0.168	0.379	0.143	0.107	0.057	0.361	0.526
6	0.170	0.169	0.378	0.140	0.107	0.057	0.361	0.524
1	0.159	0.150	0.379	0.138	0.105	0.057	0.362	0.535
2	0.155	0.141	0.373	0.137	0.104	0.056	0.346	0.529
8	0.127	0.109	0.251	0.086	0.074	0.045	0.292	0.468
7	0.119	0.109	0.242	0.101	0.077	0.046	0.285	0.457
10	0.109	0.078	0.219	0.075	0.064	0.044	0.266	0.455

- *Error analysis*

We applied various statistical and linguistic features to prioritize the abstracts with relationships between target kinase and DIS/BP mentions. However, this task is very challenging by nature. First, there are no "non-curatable" or negative documents provided in the training set. Without such negative cases, most classification methods are difficult to distinguish the curatable versus non-curatable documents. Second, the relationships between DIS/BP mentions and the target kinases are not clearly curated in the training set. For example, one abstract may have various DIS/BP mentions. Therefore, it is difficult to find the correct triples for feature extraction in our method. Third, the low recognition rate of GNormPlus missed about 20~40% of the kinases (60% of the missed kinases are not in the abstracts). Thus, about 20~40% of the relevant documents would not be found in our results. Finally, according to the official results, our best performance acquired a P100 = 0.052, R100 = 0.327 in kinases/diseases and P100 = 0.065, R100 = 0.563 in kinases/biological processes. The average number of curatable articles are 15.90 and 11.55, respectively. For each kinase with an axis, the average number of curatable articles among the entire PubMed database is extremely low. Thus, we suspect that human curators may only include articles with evidence within the experimental results section. Therefore, those excluded articles cannot be filtered by our methods which use only titles and abstracts. According to the above reasons, this task is much difficult than many other traditional document classification tasks.

CONCLUSION

In summary, we used several machine learning methods with frequency, location, and NLP features for the neXtProt triage task, which aims to specifically retrieve PubMed articles with biomedical relations among kinases, diseases, and biological processes. The average numbers of curatable articles in the testing set are rare (15.90 of target kinases/diseases articles and 11.55 of target kinases/biological processes articles). Thus, the biocurators using our methods can retrieve 32.7% (5.2 articles) and 56.3% (6.5 articles) of curatable articles among all PubMed articles with only reviewing 100 articles returned by our best method. Therefore, we believe our method can effectively accelerate the manual curation efforts used today. In our future work, we plan to examine the triage task based on full texts, and investigate a robust machine learning approach which is capable of using curatable labeled data only.

ACKNOWLEDGMENT

The authors would like to thank the organizers of the BioCreative VI Track 2. We also thank Shazia Dharssi for proofreading the manuscript and the authors of the individual open source systems used in this work. This research is funded by the National Institutes of Health, National Library of Medicine.

REFERENCES

1. Mottin, L., et al., *neXtA5: accelerating annotation of articles via automated approaches in neXtProt.* Database (Oxford), 2016. **2016**.
2. Poux, S., et al., *On expert curation and scalability:UniProtKB/Swiss-Prot as a case study.* Bioinformatics, 2017.
3. Leaman, R. and Z. Lu, *TaggerOne: joint named entity recognition and normalization with semi-Markov Models.* Bioinformatics, 2016. **32**(18): p. 2839-46.
4. Wei, C.-H., H.-Y. Kao, and Z. lu, *GNormPlus: An Integrative Approach for Tagging Genes, Gene Families, and Protein Domains*. Vol. 2015. 2015. 918710.
5. Friedman, J.H., T. Hastie, and R. Tibshirani, *Regularization Paths for Generalized Linear Models via Coordinate Descent.* 2010, 2010. **33**(1): p. 22.

6. Smith, L.H., W. Kim, and W.J. Wilbur, *PROBE: Periodic Random Orbiter Algorithm for Machine Learning*. 2012.
7. Zhang, Y. and B. Wallace, *A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification*. 2015.
8. Meyer, D., *Support Vector Machines The Interface to libsvm in package e1071*. Vol. 1. 2001.
9. Pennington, J., R. Socher, and C. Manning. *Glove: Global Vectors for Word Representation*. in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2014. Association for Computational Linguistics.
10. Wei, C.H., et al., *tmVar: a text mining approach for extracting sequence variants in biomedical literature.* Bioinformatics, 2013. **29**(11): p. 1433-9.
11. Wei, C.H., et al., *tmVar 2.0: Integrating genomic variant information from literature with dbSNP and ClinVar for precision medicine.* Bioinformatics, 2017.
12. Cer, D., et al., *Parsing to Stanford Dependencies: Trade-offs between Speed and Accuracy*. 2010.

KinDER: A Biocuration Tool for Extracting Kinase Knowledge from Biomedical Literature

Daniel Dopp[1*], Adam Morrone[2*], Indika Kahanda[3†]
Affiliation: [1]University of Kentucky, [2]Liberty University, [3]Montana State University
*equal contribution, [†]corresponding author

Abstract—Kinases are enzymes that mediate phosphate transfer. Extracting information on kinases from biomedical literature is an important task which has direct implications for applications such as drug design. In this work, we develop KinDER, Kinase Document Extractor and Ranker, a biomedical natural language processing tool for extracting functional and disease related information on kinases. This tool combines information retrieval and machine learning techniques to automatically extract information about protein kinases. First, it uses several bio-ontologies to retrieve documents related to kinases and then uses a supervised classification model to rank them according to their relevance. This was developed to participate in the *Text-mining services for Human Kinome Curation* Track of the BioCreative VI challenge. According to the official BioCreative evaluation results, KinDER provides state-of-the-art performance for extracting functional information on kinases from abstracts.

Keywords—kinase; proteins; machine learning; biomedical natural language processing; BioCreative; text classification; supervised learning

I. Introduction

With the steady advancement of computing power and decline in memory cost over the years has come the ability to work with increasingly larger data sets in more complex ways. These opportunities have opened up the relatively new fields of computer-driven bioinformatics and natural language processing. These two areas, when in conjunction, can allow for automatic extraction of important information from biomedical literature written in plain, unstructured text. An example of where this is advantageous could be having the ability to intelligently search through all existing journal articles about a specific cellular structure in order to aggregate current knowledge about that structure. This process is currently done by hand via human curators. As there are literally millions of journal articles published each year, there is much room for improvement. One such group of bio-entities of high interest are the human protein kinases, a specific type of enzyme that can phosphorylate (add a phosphate to) other proteins. This process can activate or inhibit various other proteins, and plays an important role in cellular communication and hormone action (1). An automated, intelligent search tool for protein kinases could dramatically improve the curation process and potentially assist the scientific community in better understanding these important proteins. This report describes the development, implementation, and testing of a pipeline to do just this. In particular, the KinDER (Kinase Document Extractor and Ranker) pipeline allows users to enter a specific human protein kinase in addition to an axis (be it disease, or biological function), and returns predictions of which documents from either a collection of PubMed database journal articles or MEDLINE database abstracts contain relevant information to those criteria. Furthermore, KinDER can be used to predict ~500 character snippets of text which contain relevant information to the search criteria. This tool was developed in order to participate in the Track 2 (Text-mining services for Human Kinome Curation) of BioCreAtIvE (Critical Assessment of Information Extraction systems in Biology) challenge held in 2017.

II. Methodology

KinDER is composed of two main components: 1) Document Retrieval component which retrieves documents annotated with kinases and axis terms using dictionaries and 2) Document Ranking and Information Extraction component which uses machine learning to rank those documents based on relevancy, as depicted in the figure below. There are a significant number of data processing steps that occur inside these two components that make up the full KinDER pipeline (Fig. 1). The following subsections will describe those steps in more detail.

A. Data

We use the BioCreative Track 2 official dataset as the input data for KinDER. Included in this were PubMed articles (approx. 260,000) and MEDLINE abstracts (approx. 4.4 million) in BioC format (2), lists of kinase names and synonyms, and a gold standard dataset of kinase names and associated relevant documents. This challenge has three subtasks: Abstract Triage, FullText Triage, and Snippet Selection. In order to annotate documents based on their relevancy to the disease axis (DIS), we considered the HPO (3), ORDO (4), NCITd (5) (hand culled subset comprised of only disease related subsections of NCIT), PDO (6), OAE (7), IDO (8), ICD10 (9), MeSH (10) and DOID (11) bio-ontology annotation dictionaries available from the NCBO annotator website (12). For annotating the biological process axis (BP) we considered the GO (13) dictionary from NCBO as well as a concept recognition dictionary developed by Funk et al. (14-

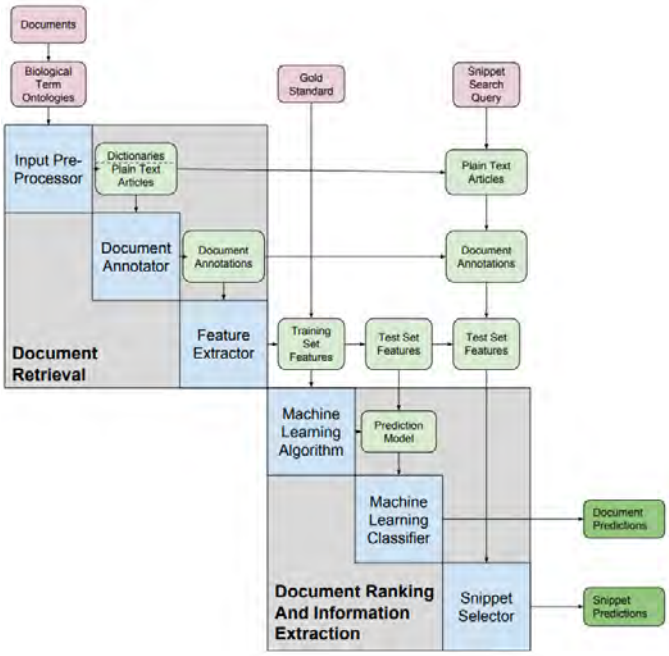


Fig. 1. N2 (N-squared) Diagram of KinDER Pipeline.

15), which we refer to as GO2. According to our preliminary results (see Fig. 2 and Fig. 3), we chose HPO and NCITd for DIS annotation and GO for BP annotation.

B. *Input Pre-Processing*

This first stage takes the input data described previously and converts it into formats useful for the annotation stage. The document annotator chosen for this pipeline was ConceptMapper (CM) (16), an annotation engine for the Apache UIMA framework (17). This tool is discussed in more detail in the next subsection. Input processing has two main steps: article extraction and dictionary creation. In order to handle the large collection of documents provided in BioC format, a custom python library was developed using lxml's iterparse object (18), which significantly improves memory efficiency compared to existing libraries. This step also consisted of writing scripts which converted BioCreative's XML lists of kinase synonyms, NCBO's CSV dictionaries, and the GO.obo dictionaries into XML formatting for ConceptMapper. Original BioCreative kinase dictionaries were also enhanced by using kinase synonyms from UniProt (19), adding any new synonyms that were not already provided. These appended dictionaries were then run through string-processing scripts to convert Roman numerals to Arabic numerals and remove unnecessary spaces between words. These variants were added as additional synonyms, as opposed to replacing old ones.

C. *Document Annotation*

As mentioned previously, CM was chosen as the primary dictionary look up tool for the KinDER pipeline. This tool takes a directory of text files to annotate, as well as a dictionary to annotate with. By default, it automatically handles all stemming and stop word removal. For improving efficiency, the CM output files were compressed into custom summary structs containing basic information on each annotation made including the term that was a "hit" as well as its position in the document and the canonical term which it refers to. In addition, these structs stored metadata about the documents such as the number of total hits, the number of unique terms, and sets/counts of matched terms.

D. *Feature Extraction*

The main goal of this stage is to provide meaningful information to enable successful downstream classification of documents as relevant or irrelevant. This task is broken down into three main processes: 1) cross-reference validation, which attempts to filter out obviously irrelevant documents (i.e. documents that do not contain both a kinase and an axis term), 2) feature vector generation, which creates vectors of meaningful metadata that the downstream machine learning algorithm can use to learn from, and finally 3) creation of corresponding binary labels for the training subset of feature vectors based on the BioCreative gold standard.

In generating feature vectors to train the machine learning based Document Ranking portion of the KinDER pipeline, two types of features were generated, both using the set of documents that made it through the initial round of cross referencing based selection. The first approach is the standard Bag of Words (BOW) feature model which uses TFIDF features values (20). In our model, two and three-gram term combinations were also included in the vocabulary.

For the second approach, six metrics were chosen as features (referred to as the "engineered feature set" or ENG). They are, Kinase Score: the number of kinase annotations normalized by total words, Axis Score: the number of axis term annotations normalized by total words, Relevancy Score: The product of the kinase score and axis score, Proximity Score: The minimum number of words separating a kinase and axis annotation, and Proximity 10-Count and Proximity 50-Count: the number of pairs of kinase and axis annotations that are within 10 and 50 words of one another. We apply standard pre- and post-processing techniques including stemming, and the removal of standard English stop words before constructing both types of above features.

E. *Machine Learning Model Selection and Training*

We model this problem as a binary classification problem in which we distinguish between *relevant* vs *irrelevant* articles. We used the Scikit-learn (21) Python machine learning library for implementing the machine learning models. An initial model selection phase was conducted comparing three supervised classification algorithms and it was determined that Support Vector Machines (SVMs) were the most promising avenue. For the BP FullText and BP Abstract subtasks, eight SVM models were evaluated based on SVM kernel (linear vs gaussian) and feature type (BOW vs ENG). For the DIS FullText and DIS Abstract subtasks, sixteen models were

evaluated based on kernel (linear vs gaussian), features (BOW vs ENG) and the ontology (HPO vs NCITd).

Each classifier model was trained using the full set of gold standard relevant documents and only a 10-20% random sample of the total irrelevant documents. Several classifiers utilizing BOW were also restricted to feature vectors of total length 100,000 (only the 100,000 most common terms). It was found through preliminary experiments that restrictions on training set size and feature vector length did not significantly impact model scoring (data not shown). Computational efforts were performed on the Hyalite High-Performance Computing System, which is operated and supported by University Information Technology Research Cyberinfrastructure at Montana State University.

F. Test Kinase Classification and Ranking Paradigm

Saved classification models were used for ranking the documents based on their relevance. Document subsets for the test data created in the document annotation stage were fed into their respective classifiers and assigned a classification and confidence score. All documents within a subset were sorted based on the classifier confidence score.

G. Snippet Selection

In order to extract a snippet of text 500 characters or less which contained sufficient relevant information to make an accurate annotation for the article, we used the following method. First, the two annotated terms, kinase and axis, that were in closest proximity in the article was identified. Next an approximate 500-character excerpt encapsulating the two terms as close to the middle as possible was captured. Finally, the excerpt length was rounded down in order to begin and end at the start and end of sentences.

III. Experimental Setup

A. Document Retreival

To determine the best ontology for document retrieval, standard metrics of precision, recall, and F-1 Score were utilized. To calculate these, two sets of articles of equal size, one containing gold standard positives and the other containing gold standard negatives were created. The default settings of ConceptMapper were used.

B. Document Classification and Ranking

Three classifier models were compared: K-Nearest Neighbors, Support Vector Machines, and Naïve Bayes. For evaluating the machine learning models, in both comparing models in the initial model selection phase and selecting hyperparameters when tuning models for the ranking phase, a 3-fold stratified cross validation technique (22) was used. We used AUROC (23) as our evaluation measure. In training models for ranking, a grid search with nested cross validation (24) approach was used.

IV. Results

A. Document Retrieval

According to the results of ontology comparison for both the Abstract and the FullText sets depicted in Fig. 2 and Fig. 3, we chose NCITd and HPO for the disease axis, and GO for the function axis.

B. Ranking and Information Extraction

Initial SVM scoring results were promising for the BOW model which significantly outscored the engineered feature set as seen in Fig. 4 and Fig. 5. Furthermore, the linear kernel SVM performed best across all subtasks and ontologies, slightly beating out gaussian kernel models, likely due to a larger hyperparameter search space used because of more efficient training times.

C. Official BioCreative Track 2 Results

In addition to prediction made by our machine learning models described above, we made a set of submissions based on several *rule-based* models that were each using the six ENG feature types. In this method, each document was assigned a relevancy score or an aggregate score of all calculated feature vectors, and the predictions were made purely on this basis without any machine learning. According to Table I which shows the MAP (mean average precision –

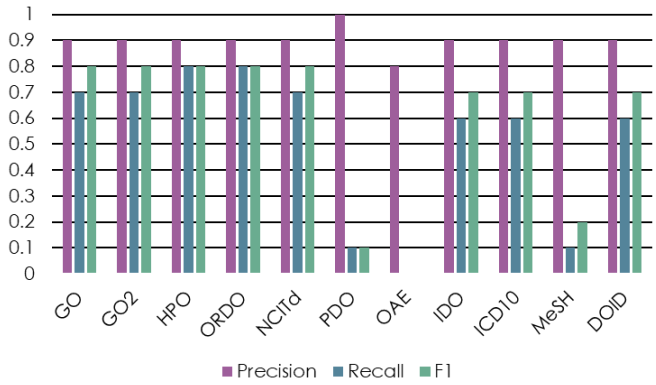


Fig. 2. Bio-ontology comparison for Abstract subtask.

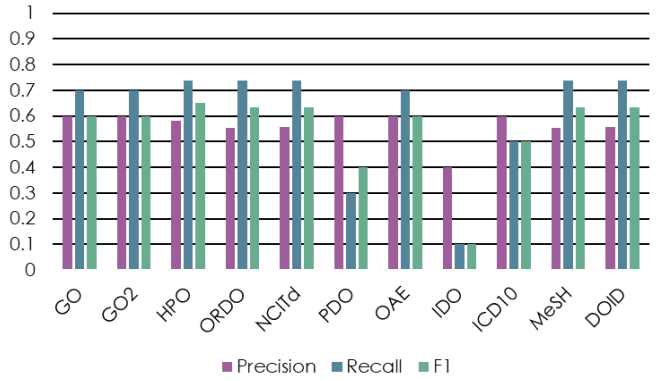


Fig. 3. Bio-ontology comparison for FullText subtask.

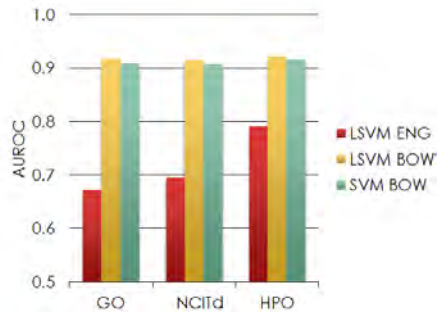


Fig. 4. SVM performance on Abstract subtask. LSVM/ SVM: SVMs using linear/gaussian kernel, BOW: bag-of-words features, ENG: engineered features.

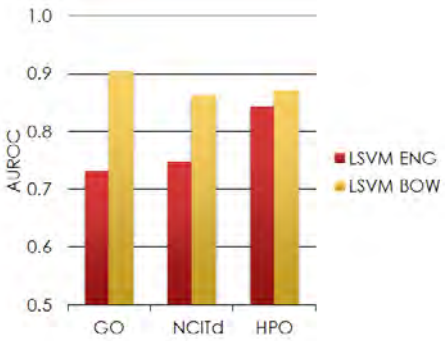


Fig. 5. SVM performance on FullText subtask. LSVM/ SVM: SVMs using linear/gaussian kernel, BOW: bag-of-words features, ENG: engineered features.

the higher is better) scores for our top three submissions in each of the subtasks, and as predicted by our preliminary tests, using a BOW machine learning model outperformed engineered feature sets. In addition, however, we observe that the rule-based methods using engineered features outperformed the machine learning methods for subtask 2 (FullText). It is important to note that, for the Abstract BP subtask, KinDER provides state-of-the-art performance among all submissions.

V. Conclusion and Future Work

To conclude, KinDER has been shown to have the potential to become an effective tool for automating biocuration efforts, particularly in the functional domain. However, much work is still necessary to improve both the recall of document annotation and the ranking paradigms.

The creation and experimentation with KinDER revealed several additional avenues of necessary research in the field. Perhaps the most glaring problem was the lack of a comprehensive list of synonyms for proteins kinases. Though several synonym lists exist, we found through experimentation that none are exhaustive when it comes to the various ways that authors notate kinases. We also found that manually expanding the synonym lists (for instance changing roman numerals to numbers and vice versa) increased our recall. A more comprehensive list would improve results for computer-driven biocuration.

TABLE I. Official BioCreative results.

Subtask	Model	MAP
Abstract - DIS	HPO - LSVM BOW	0.098
	NCITd - LSVM BOW	0.096
	NCITd - Relevancy Score[a]	0.080
Abstract - BP	GO - LSVM BOW	0.201
	GO - Relevancy Score[a]	0.197
	GO – LSVM BOW[b]	0.187
FullText - DIS	NCITd - Aggregate Score[a]	0.118
	NCITd - Relevancy Score[a]	0.112
	HPO - Relevancy Score[a]	0.100
FullText - BP	GO - Aggregate Score[a]	0.293
	GO - Kinase Score[a]	0.278
	GO - Proximity Score[a]	0.271

a. Rule-based methods.

b. Using only 20% of the training data to train the classifier.

Within the machine learning portion of our work, many improvements could be made in the comprehensiveness of model selection and training. The extent of model selection was fairly limited due to resource constraints for this study and examining further models e.g. random forests or neural networks may lead to improved predictions over SVMs. Training SVMs is a highly resource intensive process for larger datasets, making it difficult to test more than a handful of hyperparameters. A more extensive parameter sweep trained on a larger, more balanced dataset would likely improve KinDER's performance. In addition, should a golden standard data set containing examples for every kinase we are interested in ranking be released this problem could be rethought as a multiclass classification problem which would simplify many aspects of the problem.

In addition to triage improvements, our process for snippet selection did not incorporate any machine learning techniques. If we were able to incorporate ML, we believe our snippet selection process would improve as well. Lastly, though KinDER is a fully functional standalone pipeline, its current web user interface is very limited. The evolution of KinDER into an end-to-end tool for biocuration could lend itself well to future bio-curation projects.

Acknowledgement

The authors would like to thank Karen Stengel of Montana State University for her input as a domain expert for improving Kinase dictionaries. We would also like to thank the National Science Foundation for funding the REU (Research Experience for Undergraduates) program at Montana State University within which this research was conducted.

References

1. Marieb,E., Hoehn,K., (2013) The Endocrine System. *Human Anatomy & Physiology,* **9**, 606-619.

2. Comeau,D., Doğan,R., Ciccarese et al. (2013) BioC: A minimalist approach to interoperability for biomedical text processing. *Database*, 15.

3. Robinson,P., Köhler,S., Bauer et al. (2008) The Human Phenotype Ontology: A Tool for Annotating and Analyzing Human Hereditary Disease. *The American Journal of Human Genetics*, **83**, 610-615.

4. Vasent, D., Chanas,L Malone et al. (2014) ORDO: An Ontology Connecting Rare Disease, Epidemiology and Genetic Data. *Phenotype data at ISMB2014*.

5. Sioutos,N., Coronado,S., Haber et al. (2007) NCI Thesaurus: A semantic model integrating cancer-related clinical and molecular information. *Journal of Biomedical Informatics*, **40**, 30-43.

6. (2016) Pathogenic Disease Ontology *http://bioportal.bioontology.org/ontologies/PDO*

7. He,Y., Sarntivijai, S., Lin et al. (2014) OAE: The Ontology of Adverse Events. *Journal of Biomedical Semantics*, **5**, 29.

8. Cowell,L., Smith,B. (2011) Infectious Disease Ontology. *Infectious Disease Informatics,* **27**, 373-396.

9. National Center for Health Statistics. (2016) *https://www.cdc.gov/nchs/icd/icd10.htm*

10. U.S. National Library of Medicine. Medical Subject Headings – Home Page. *https://www.nlm.nih.gov/mesh/*

11. Schriml,L., Arze,C., Nadendla et al. (2012) Disease ontology: A backbone for disease semantic integration. *Nucleic Acids Research*, **40**.

12. Jonquet,C., Shah,N., Cherie et al. (2009) NCBO Annotator: Semantic Annotation of Biomedical Data. *Iswc,* 2-3.

13. Ashburner,M., Ball,C., Blake et al. (2000) Gene Ontology: tool for the unification of biology. *Nature Genetics*, **25**, 25-29.

14. Funk,C., Baumgartner,W., Garcia et al. (2014) Large-scale biomedical concept recognition: an evaluation of current automatic annotators and their parameters. BMC Bioinformatics, **15**, 59.

15. Funk,C., Kahanda,I., Ben-Hur et al. (2015) Evaluating a variety of text-mined features for automatic protein function prediction with GOstruct. *Jounral of biomedical semantics*, **6**, 9.

16. Tanenblatt,M., Coden,A., Sominsky, I. (2010) The ConceptMapper Approach to Named Entity Recognition. *Proceedings of the Seventh conference on International Language Resources and Evaluation LREC10,* 546-551.

17. ASF (2013) Apache UIMA. *http://uima.apache.org/*

18. Richter,S. (2015) lxml - XML and HTML with Python. http://lxml.de/.

19. Bateman,A., Martin,M., O'Donovan et al. (2017) UniProt: The universal protein knowledgebase. *Nucleic Acids Research*, **45**, D158-D169.

20. Manning,C., Raghavan,P., Shutze,H. (2009) An Introduction to Information Retrieval. 569.

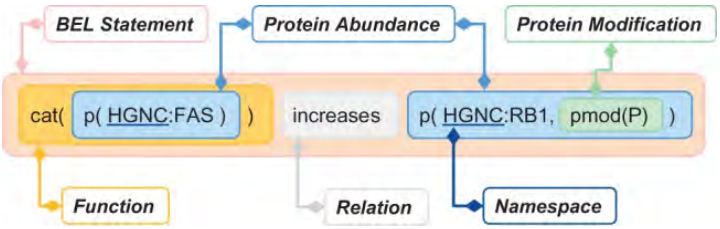
21. Pedragosa,F., Varoquaux,G., Gramfote et al. (2012) Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research,* **12**, 2825-2830.

22. Witten,I., Frank,E., Hall, M. (2016) Data Mining: Practical Machine Learning Tools and Techniques 3rd Edition.

23. Bewick,V., Cheek,L., Ball,J. Statistics review 13: Receiver operating characteristic curves. *Critical Care*, **8**, 508-512.

24. Cawley,G., Talbot,N. (2010) On Over-fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation. *Journal of Machine Learning Research*, **11**, 2079-2107.

Track 3
Extraction of causal network information using BEL



Overview of BEL Track: Extraction of Complex Relationships and their Conversion to BEL

Sumit Madan[1], Justyna Szostak[2], Jens Dörpinghaus[1], Julia Hoeng[2] and Juliane Fluck[1]
[1]Fraunhofer Institute SCAI, Schloss Birlinghoven, 53757 Sankt Augustin, Germany
[2]Philipp Morris International R&D, Philip Morris Products S.A, Quai Jeanrenaud 5, 2000 Neuchatel, Switzerland

***Abstract*— Biological signaling is complex and our knowledge about it is often only available in literature. Signaling can involve small molecules as well as proteins that can be activated or deactivated by various regulations such as ligand binding, complex forming, modification status or miRNA binding. Changes in signaling influence biological processes and/or are involved in disease etiology. The Biological Expression Language (BEL) has been created to store this kind of information in a structured form that can be used for network generation and visualization as well as interpretation of experimentally generated data. The BioCreative VI BEL track provides training data and an evaluation environment to encourage the text mining community to tackle the automatic extraction of such complex relationships as well as converting it to BEL. Although only a few groups participated in this track, the groups participating the second time could drastically increase their performance. The best system reached 32% F-score for extraction of complete BEL statements (task 1) and, when given the named entities, above 49%. Beside rule-based systems, methods involving hierarchical sequence labeling and neural networks are adapted to this task. For the second task in the BEL track, finding evidence text snippets for a given statement, despite the provided training data, only one team took part.**

Keywords—text mining; relation extraction; named entity recognition; entity normalization; evaluation; BEL

I. Introduction

Biological Expression Language (BEL) has been introduced to allow the formalization of causal biomedical relationships (1). The resulting BEL statements can be assembled into causal networks, which can be easily queried or used for data mining. The manual extraction of this information from literature is time consuming. Consequently, there is a high demand for automatic support. Currently, the automatic extraction has not yet reached the performance to allow for fully automatic extraction of this information. Even for humans, depending on training and application area, formalizing complex knowledge is demanding and not unambiguous. Nevertheless, due to the increase of large scale experiments and the requirements of molecular knowledge in precision medicine, the demand for structured biomedical knowledge is increasing. The BEL track has been introduced in BioCreative V 2015 to provide training data for the extraction and BEL translation of complex relationships over a set of different named entity classes (2). In addition to the relationship information, BEL has the advantage to provide provenance information in form of text evidences that are well suited to serve as training set. During the first BEL track, we appraised a large corpus of BEL documents resulting in a training corpus of BEL statement–sentences pairs (3). A development and the test set were analyzed by annotators and curated to assure that all possible BEL statements are associated with the sentences. In addition, information resources to explain BEL and an evaluation framework has been set up (2). For the current task, novel and yet unpublished test data from the disease context of Ulcerative Colitis (4,5) has been created. In the next chapter, the task, relevant resources, the created test set and result summaries of the participating systems are presented.

II. Task Overview

A. Biological Expression Language and used namespaces

BEL statements encode semantic triples with subject, relationship and object. An example BEL statement with the corresponding sentence is shown in Fig. 1. For the BEL track, we focus on two causal relationship types: increase and decrease. Subject and object contain entities that are normalized to so-called namespaces. Those namespaces are generated from database entries (e.g. human genes from HGNC[1], mouse genes from MGI[2] and chemical entities from ChEBI [3] database). Other namespace origin either from ontologies, such as the biological processes subtree of the Gene Ontology for GOBP[4], or from terminologies, such as the diseases namespace MESHD [5] from the Medical Subject Heading terminology disease subtree. By using the normalized entities from such namespaces, the resulting statements can be integrated and merged to networks as well as aligned to other data.

For the different entities, different class abundances are assigned: the abundance function *a()* is assigned to chemicals, *bp()* for biological processes and *path()* (pathology) for diseases. For genes, different abundances *g()* (gene), *r()*

[1] HGNC stands for HUGO Gene Nomenclature Committee (http://www.genenames.org)
[2] MGI stands for Mouse Genome Informatics (http://www.informatics.jax.org/)
[3] ChEBI stands for Chemical Entities of Biological Interest (https://www.ebi.ac.uk/chebi/)
[4] GOBP stands for Gene Ontology Biological Process
[5] MESH disease subtree is available at https://meshb.nlm.nih.gov/treeView.

We acknowledge support of our research from Philip Morris International.

(mRNA) or *p()* (protein) are possible, but only *p()* is used in the BEL track to reduce the complexity. In addition to those class abundances, different functions can also be assigned to the biological entities. For the BEL track, we focus on the protein phosphorylation function *pmod()*, the activity function *act()* to describe a protein activation, the *tloc()* function to describe a translocation, *deg()* for degradation and finally *complex()* to describe protein complexes. For a more detailed description we refer to Rinaldi et al. (2).

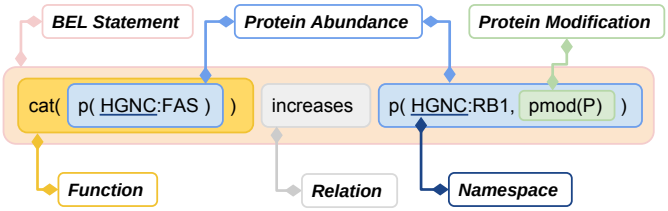


Fig. 1 Example of a BEL statement extracted manually for the sentence "Fas stimulation of Jurkat cells is known to induce p38 kinase and we find a pronounced increase in Rb phosphorylation within 30 min of Fas stimulation" (PMID:10075927). Reprinted with permission from Rinaldi et al. (2).

B. *Task description and evaluation*

The BEL track challenge is organized into two tasks evaluating the complementary aspects of the problem:

- ***Task 1: Given textual evidence for a BEL statement, generate the corresponding BEL statement.***
- ***Task 2: Given a BEL statement, provide at most 10 additional evidence sentences.***

Extraction of relationships and their coding in BEL is a complex task due to the different entity, relationship and function types. Therefore, we simplified the challenge further and provided a cascade model for evaluation of task 1. A detailed overview of all simplifications is provided online[6]. In short, HGNC or EntrezGene identifiers are accepted for the same statement and mouse orthologous identifiers are accepted as well. For the abundance function of those namespaces, all correct abundance are accepted. Furthermore, for the modification function *pmod()* and the translocation function *tloc()*, the number of arguments are reduced.

The cascade evaluation model is described in detail in Rinaldi et al. (2). Only syntactically correct statements in a correct format are accepted, but different levels of performance are evaluated. A submitted full BEL statement is automatically cut into its fragments to ensure evaluations on lower levels. On term level, only the correctness of BEL terms is assessed. On this level, the correctness of the discovered entities, the correctness of associated namespaces and their format as well as the correctness of the associated abundance/process function is measured.

On function level the correctness of discovered function is evaluated. Functions are only accepted together with their argument, the BEL term. As simplification, a complex function is only valid if at least one of its arguments is correct. On the secondary function level, the correctness of a function alone was measured, regardless of the correctness of their term-arguments.

In the relationship-level evaluation, only the entities and the relationships are considered, functions that are part of a BEL statement are not taken into account on this level. Yet again, two levels of evaluation are considered. For a full score relationship, subject, object and the relationship type must be correct. For the secondary relationship level, partial relationships, containing two correct units out of three (subject, object and relationship type), are considered fulfilled. Finally, we evaluated how many BEL statements are entirely correct.

For task 2, up to 10 sentences for each BEL statements were accepted from the participating systems. Those statements were evaluated on two levels: In the 'fully supportive level', the sentence must contain all necessary information for a biologist to create the BEL statement. In the 'partially supportive level', the sentence is correct, when context information from the paper is taken into account. For more detailed information on the evaluation criteria, we refer to Rinaldi et al. (2).

III. Training Data and Preparation of New Test Set

The training data and test data from BioCreative V (2015) is available at the datasets page[7]. The description of the training set selection and annotation are described in detail in Fluck et al. (3). For the generation of the 2017 BEL track task 1 test set, we decided to use a real-world use case and extracted new data in the disease context of Ulcerative Colitis. For the test set, we restricted the named entity classes to those that can be normalized to the gene/protein namespaces HGNC and MGI, CHEBI for chemical names, MESH for disease names and GO for biological processes and a restricted set of relationship types and functions defined above. For the extraction, we used automatic support in form of the BELIEF workflow (6). This workflow pre-annotates and normalizes the named entities and suggests BEL statements in a user-friendly curation environment. In the frontend, the curator can browse through the text, search for unrecognized named entities and edit or add statements. Only user-selected statements are exported by the system.

In a first step, two curators independently extracted the information from the same full texts. In the comparison of results, it became clear that it is very tedious to extract all possible statements for a document and, in addition, we would create very similar statements only with different experimental settings, which are irrelevant for the task 1. It was also not feasible that independent curators select the same sentences for curation in full text. Therefore, we decided towards a more straightforward approach. One person selected the relevant sentences and extracted all BEL statements from this sentence. The second curator analyzed and edited this set. Finally, differences were discussed in an annotation jamboree. In Table I, an overview about the number of different statements, articles, entities and relationship types and functions of the task

[6] http://wiki.openbel.org/display/BIOC/All+Functions+Evaluation+Overview

[7] https://wiki.openbel.org/display/BIOC/Datasets

1 test set are given. For task 2 BEL statements were curated from abstracts to make sure that at least one sentence could be found for every BEL statement.

TABLE I. DISTRIBUTION OF TERM, FUNCTION AND RELATIONSHIP TYPES IN THE TRAINING AND TEST SETS

Type	*Training*	*Test 2015*	*Test 2017*
Terms			
p()	19.918	346	328
a()	1.927	37	52
bp()	877	31	23
path()	244	15	2
Functions			
act()	6.332	36	79
pmod()	1.411	9	36
complex()	750	15	5
tloc()	406	13	10
deg()	205	6	4
sub()	23	0	0
trunc()	6	0	0
Relations			
increases	8.112	155	130
decreases	2.956	53	68

IV. SUPPORTING RESOURCES

The participants were provided with a range of supporting resources and a comprehensive documentation[8], containing a description of the format and detailed explanation of the evaluation process. The evaluation on the different levels of a single BEL statement was illustrated using a set of concrete example submissions as reference. Additionally, a validation and an evaluation interface[9] was provided for the participants to validate and test their generated statements during the development phase. The BEL statement validator checks the user provided BEL statements with respect to formal correctness. It provides specific error messages for invalid BEL statements. For sample, training and 2015 test set, the evaluation interface evaluates the input BEL statements based on the evaluation criteria such as term, function, relationship and full statement level. An example of a candidate evaluation is shown in Figure 2.

Further supporting resources included the BEL statements from the training, sample and 2015 test set in BioC format, which we generated automatically using a converter based on the official ruby-based BEL parser[10] and an open-source BioC ruby module [11]. A tab-separated format that contains all fragments of the BEL statements (terms, functions and relations) was automatically generated from the sample, training and 2015 test set, using the same BEL parser mentioned above. These were provided to the participants as supporting material (c.f. (2)).

For task 1 stage 1, we also provided the normalized names of all biological processes that occur in the test set as extracting such concepts is still a non-trivial task. Finally, for task 1 stage 2, we provided a file with entity information with offsets and the associated normalized concept with the namespace.

```
Sent.-Id:10004582   PMID:15909112

Sentence: In the present study, we found
that transgenic mice overexpressing wild-
type human APP gene (hAPP/+) displayed a
much higher expression of FAS, one of the
death receptor subfamily.

BEL statements
---------------------------
Gold standard BEL statements
p(HGNC:APP) -> p(HGNC:FAS)
----------------------------------------------
Prediction BEL statements
act(p(HGNC:APP)) -> bp(GOBP:"gene expression")
act(p(HGNC:APP)) -> act(p(HGNC:FAS))

Sentence based evaluation
Class          | TP | FP | FN | R      | P      | F-score
---------------| -- | -- | -- | ------ | -------| -------
Term (T)       |  2 |  1 |  0 | 100.00 |  66.67 |   80.00
Func-Sec (FS)  |  0 |  1 |  0 |      0 |      0 |       0
Function (F)   |  0 |  2 |  0 |      0 |      0 |       0
Rela-Sec (RS)  |  1 |  0 |  0 | 100.00 | 100.00 |  100.00
Relation (R)   |  1 |  1 |  0 | 100.00 |  50.00 |   66.67
Statement (S)  |  0 |  2 |  1 |      0 |      0 |       0
```

Fig. 2 An example of a candidate evaluation. The example shows the candidate sentence, the gold standard and predicted statements. For all primary and secondary levels the scores are provided (2). Abbreviations: PMID (PubMed identifier), true positive (TP), false positive (FP), false negative (FN), recall (R), precision (P). Adapted and reprinted with permission from Fluck et al. (7).

V. RESULTS

A. *Task 1: Given textual evidence for a BEL statement, generate the corresponding BEL statement.*

Four teams contributed results of their information extraction systems for task 1. A maximum number of three submissions was permitted. Table II shows the results for the task 1 in stage 1. Here the teams had to provide their own term recognition. The results are color-coded in shades of green according to the values of F-score (F), the main evaluation criterion, and supplemented by the values for precision (P) and recall (R). The best results for each evaluation metrics are marked up in bold script. In general, all teams took part on all structural levels except team 390, which excluded the function level. Team 379 and 411 already took part in the BioCreative V BEL track task 1 (2015).

For the full statement level, the best system with the team id 379 achieved an F-measure of 32%. This illustrates the difficulty of this highly structured prediction task. The teams 411 and 390 had a similar performance, although their results were quite different on other evaluation levels, e.g. the term level.

[8] https://wiki.openbel.org/display/BIOC/BioCreative+BEL+Task+Challenges
[9] http://bio-eval.scai.fraunhofer.de/cgi-bin/General_server.rc
[10] https://github.com/OpenBEL/bel.rb
[11] https://github.com/dongseop/simple_bioc

TABLE II. EVALUATION OF STAGE 1 OF TASK 1 (PREDICTION OF BEL STATEMENTS WITHOUT GOLD STANDARD ENTITIES)

Team Id	Run	Terms			Function			Function Second.			Relation			Relation Second.			Statement		
		F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R
411	r1	63.24	84.62	50.49	33.99	44.83	27.37	51.24	67.39	41.33	40.22	55.38	31.58	62.92	88.19	48.91	22.99	33.33	17.54
	r2	57.75	81.93	44.59	31.08	43.4	24.21	38.67	69.05	38.67	36.78	53.33	28.07	57.73	86.84	43.23	20.71	31.82	15.35
	r3	61.24	88.27	46.89	32.88	47.06	25.26	46.15	64.29	36	37.43	56.14	28.07	62.03	92.24	46.72	21.15	33.98	15.35
380	r1	50.88	76.82	38.03	6	60	3.16	7.5	60	4	16.77	31.71	11.4	45.14	80	31.44	7.38	15.71	4.82
	r2	55.29	81.01	41.97	6.06	75	3.16	7.59	75	4	21.52	38.64	14.91	51.06	84	36.68	10.67	22.22	7.02
	r3	67.83	72.22	63.93	20.17	50	12.63	31.25	71.43	20	24.69	28.25	21.93	62.25	70.95	55.46	10.44	12.9	8.77
379	r1	74.14	78.18	70.49	**40.54**	56.6	31.58	**55.28**	70.83	45.33	43.65	51.81	37.72	86.17	89.62	82.97	32.28	40.67	26.75
	r2	72.89	78.71	67.87	40.29	63.64	29.47	54.39	79.49	41.33	**43.77**	52.12	37.72	**86.71**	93	81.22	**32.45**	41.22	26.75
390	r1	**76.39**	81.18	72.13	0	0	0	0	0	0	29.87	25.55	35.96	65.19	60.45	70.74	18.08	16.1	20.61
	r2	**76.39**	81.18	72.13	0	0	0	0	0	0	28.92	24.19	35.96	65.23	59.29	72.49	17.88	15.53	21.05

TABLE III. EVALUATION OF STAGE 2 OF TASK 1 (PREDICTION OF BEL STATEMENTS WITH GOLD STANDARD ENTITIES)

Team Id	Run	Terms			Function			Function Second.			Relation			Relation Second.			Statement		
		F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R
411	r1	83.93	99.11	72.79	36.36	47.46	29.47	46.77	59.18	38.67	57.22	73.29	46.93	83.33	98.8	72.05	31.30	46.15	23.68
	r2	86.09	99.15	76.07	40.51	50.79	33.68	51.16	61.11	44	56.08	70.67	46.49	83.92	98.82	72.93	30.95	44.63	23.68
	r3	85.45	99.13	75.08	39.24	49.21	32.63	50	60.38	42.67	57.6	73.47	47.37	83.63	98.81	72.49	31.79	46.61	24.12
380	r1	90.20	98.83	82.95	12.39	38.89	7.37	21.74	58.82	13.33	42.52	52.94	35.53	84.24	96.61	74.67	22.66	32	17.54
379	r1	87.65	90.56	84.92	51.75	77.08	38.95	57.63	79.07	45.33	**66.83**	76.7	59.21	**92.06**	95.75	88.65	49.2	63.01	40.35
	r2	86.4	90.94	82.3	**52.55**	85.71	37.89	**58.93**	89.19	44	**66.83**	76.7	59.21	91.92	97.55	86.9	**49.6**	64.34	40.35
	r3	87.41	90.81	84.26	51.75	77.08	38.95	57.63	79.07	45.33	**66.83**	76.7	59.21	91.99	96.63	87.77	49.2	63.01	40.35
390	r1	**91.33**	99.23	84.59	0	0	0	0	0	0	43.51	41.6	45.61	86.36	90.05	82.97	23.61	25	22.37
	r2	88.36	99.18	79.67	0	0	0	0	0	0	42.47	44.29	40.79	83.41	91.19	76.86	24.06	28.07	21.05
	r3	76.71	98.96	62.62	0	0	0	0	0	0	25.68	29.38	22.81	71.76	85.98	61.57	15.06	18.47	12.72

On the secondary relation level, it is sufficient that only two out of the three components of relationship (subject, object, relation type) are correct for the statement to count as a positive match. This level is introduced in order to give credit to results, which although partially correct, could still be useful in the context of a semi-automated approach as suggestions to a human curator. Here, up to 87% F-score were achieved, which is quite impressive. Whereas on the relation level, the highest F-measure was around 43%. This shows a drop of around 40% in comparison to the secondary level.

Furthermore, the comparison of all teams shows that most of the teams (except team 390) built systems that focus more on precision rather than recall. High scores on the relation level do not necessarily correlate with high scores on the full statement level. This is due to the fact that full statement level combines all structural levels.

The results for task 1 stage 2 are shown in Table III. In this stage, the gold standard concepts together with their specific text spans were made available to the teams. All teams could significantly benefit and improve on the level of full statements, which shows the importance of high-quality term recognition for further higher-level recognition tasks. Team 379 reached the highest F-score of 49.6% with the provided terms. In comparison to stage1, the score was increased up to 17% on the same test set. Similar performance increases can be seen on the function and relation level, too.

B. *Task 2: Given a BEL statement, provide at most 10 additional evidence sentences.*

For this task only one team participated. The team asked the organizers whether they can choose a different setting for the submission. In agreement with the organizers, two runs with two different configurations and only 5 ranked sentences for each run were submitted. The correctness of the provided evidence sentences was evaluated manually and rated on two different levels of strictness:

1. Fully supportive: Relationship is fully expressed in the sentence.
2. Partially supportive: Relationship can be extracted from the sentence if context sentences or biological background knowledge are taken into account.

To evaluate the quality of the curation results, we calculated an inter-annotator agreement. To tackle this task part of the manual curation was carried out by two different curators. For 150 entries, we observed a high agreement of 93% (kappa statistic: 0.75) and 91% (kappa statistic: 0.79) for the categories fully and partially supportive, respectively.

TABLE IV. EVALUATION RESULTS OF TASK 2 INCLUDING MEAN AVERAGE PRECISION (MAP)

Runs	Criterion	TP	FP	Precision	MAP
Run 1	Full	117	265	30.6%	59.6%
	Partial	175	207	45.8%	77.5%
Run 2	Full	121	261	31.7%	50.2%
	Partial	192	190	50.3%	76.7%

As shown in Table IV the system provided 382 evidence sentences for 98 BEL statements in each run (mean 3.9 sentences per statement). In run 1 for 55 BEL statements, there was at least one entirely correct evidence sentence, for 71 statements at least one sentence meeting the partially supportive evaluation condition, and in run 2, 58 and 70 BEL statements satisfied the fully and partially supportive evaluation condition, respectively. Table IV also shows the detailed numbers for TP, FP and the resulting precision at the micro level. Around one third of all sentences fully expressed the desired relationship. In order to assess the ranking quality of the system, we computed the mean average precision (MAP). Although the first run has a slightly lower precision compared to the second run, the MAP is considerably higher, especially for full supportive sentences. Overall, based on the results and the low number of participants, task 2 seems to be as difficult as task 1.

ACKNOWLEDGMENT

We would like to thank Lisa Langnickel for her support in the generation of the test data and the evaluation of task 2 data.

REFERENCES

1. Slater, T. and Song, D. Saved by the BEL: ringing in a common language for the life sciences. 2012.
2. Rinaldi, F., Ellendorff, T. R., Madan, S., et al. (2016) BioCreative V track 4: a shared task for the extraction of causal network information using the Biological Expression Language, *Database*, 2016.
3. Fluck, J., Madan, S., Ansari, S., et al. (2016) Training and evaluation corpora for the extraction of causal relationships encoded in biological expression language (BEL)., *Database (Oxford).*, 2016.
4. Sartor, R. B. (2006) Mechanisms of Disease: pathogenesis of Crohn's disease and ulcerative colitis, *Nat. Clin. Pract. Gastroenterol. Hepatol.*, 3, 390–407.
5. Kaistha, A. and Levine, J. (2014) Inflammatory bowel disease: the classic gastrointestinal autoimmune disease., *Curr. Probl. Pediatr. Adolesc. Health Care*, 44, 328–34.
6. Madan, S., Hodapp, S., Senger, P., et al. (2016) The BEL information extraction workflow (BELIEF): evaluation in the BioCreative V BEL and IAT track, *Database*, 2016, baw136.
7. Fluck, J., Madan, S., Ellendorff, T. R., et al. (2015) Track 4 Overview: Extraction of Causal Network Information in Biological Expression Language (BEL), *Proc. Fifth BioCreative Chall. Eval. Work.*

BELMiner – Information extraction system to extract BEL relationships

Ravikumar Komandur Elayavilli[1], Majid Rastegar-Mojarad[1,2], Hongfang Liu[1]
Affiliation: [1]Department of Health Sciences Research, Mayo Clinic, USA
[2]University of Wisconsin-Milwaukee, Milwaukee, WI, USA

Abstract— **In this work, we describe an improved version of BELMiner that extracts Biological Expression Language(BEL) statements from evidence sentences provided for BioCreative BEL 2017 task. The current system builds upon the basic infrastructure of BELMiner; a system was developed to extract BEL relationships from evidence sentences for the BioCreative BEL 2015 task. Training a state of the art machine learning NER architecture using a pooled corpus from diverse shared tasks and marshalling their results with other existing state of the art NER tools, incorporating the latest Stanford parser 3.8 textual entailment functionalities, graph based traversal to extract relations, handling double negations are some of the new functionalities that we incorporated in BELMiner. The system achieved an overall F-measure of 49.6% with gold standard entities, while it achieved a lower performance of 32.45% with the entities extracted by an ensemble of NER systems on blind test data. For relation extraction, the system achieved an F-measure of 66.83% on a blind test data set with gold standard entities. We observe a significant improvement in the state of the art performance in BEL statement extraction by over 14% when compared to the performance of the best system on 2015 data set.**

Keywords—BEL; BELMiner; graph based relation extraction; biomedical information extraction

I. INTRODUCTION

Biomedical literature has been a rich resource for information on biological pathways. Tapping into the information in the literature on signaling pathways is of great importance that will help bridge the gap in the curation of biological pathways. One approach is to crowdsource the human curation of biological events and pathways through intuitive and effective user interfaces such as PubTator [1], BELIEF [2] and other similar tools. This curation initiative can be substantially augmented through text mining efforts.

Formal representation of textual assertions in biological literature is important to fully realize any of the curation and text mining initiatives. There has been considerable attention on Biological Expression Language(BEL) [3] by system biologists in the recent past. It is one of the suitable representation languages to formalize signaling pathways from biomedical literature. BioCreative shared task organizers organized a very important task involving formalizing the relation extracted from biomedical text in BEL framework in 2015 [4]. Multiple teams participated in the task and the state of the art performance achieved F-measure of 35% when the system used the gold standard entities. Our system BELMiner [5] achieved an F-measure of 25.6%. As a continued effort, this year additional test data was provided to validate the ability to extract BEL statements from evidence sentences. In this paper, we describe the performance of an improved version of BELMiner, which we call BELMiner 2.0 on the BioCreative 2017 BEL extraction task. In the following sections, we briefly describe the various improvements we made to the BELMiner, its performance and brief error analysis. Due to space constraints, we provide only a brief description of our system.

II. SYSTEM DESCRIPTION

Figure 1 outlines the overall architecture of BELMiner 2.0. The system consists of the following components executed in sequence to process the evidence statement, with each component incrementally contributing towards BEL statement extraction. 1) Extraction of normalized entities 2) Identify dependency structure 3) graph based traversal to extract causal relationships 4) Formalization of causal relations into BEL statements and 5) Filter out irrelevant BEL statements.

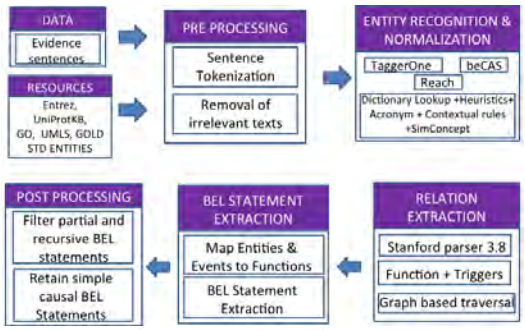


Fig. 1. BELMiner architecture.

We briefly describe the individual steps of BELMiner 2.0 below. In Figure 2 we have illustrated the important steps of BELMiner 2.0 using an example sentence.

A. Preprocessing

BELMiner 2.0 pipeline starts with some simple pre-processing such as sentence tokenization. We also have few simple regular expressions to trim unnecessary phrases like "(Fig 2A)", "Figure 1"" from the sentences to avoid errors in the subsequent steps.

B. Entity recognition and normalization

In BELMiner, we assembled an ensemble of state of the art entity normalization tools such as PubTator and beCAS [6], further supplemented by dictionary based lookup [7] to

extract normalized entities from the evidence sentences provided for the task. Subsequent analysis of the impact of different NER components showed certain gaps in the NER tools. Most of these tools are trained on the specific corpus and hence do not generalize well for the BEL NER task. In this work, we used TaggerOne [8], a trainable semi-Markov structured linear classifier for any entity types. It also uses supervised semantic indexing for entity normalization. Its unique feature is its ability to jointly model NER and normalization.

We made the following changes to the core components of the TaggerOne framework. First, we replaced the Ab3P acronym detection component with the algorithm of Schwartz and Hearst algorithm [9]. We did joint training for the gene, chemical, and disease entity detection and normalization using diverse corpus from prior BioCreative shared task data sets [10]. While the TaggerOne infrastructure had inherent support for Chemical and Disease entity normalization we did additional implementation to extend its functionality for gene normalization. We used internally generated lexicon integrated from different sources such as UniProtKB [11], Entrez Gene [12], HGNC [13] and MGI [14]. We supplemented the annotations provided by TaggerOne with annotations from other external tools such as beCAS and Reach [15]. Our strategy was similar to the one described in the earlier paper; to build consensus across multiple NER tools for the gene, chemical, and diseases. For Gene ontology term normalization we used dictionary look-up approach where we used only the terms given by the task organizers. In addition to the terms, we included their synonyms from GO ontology.

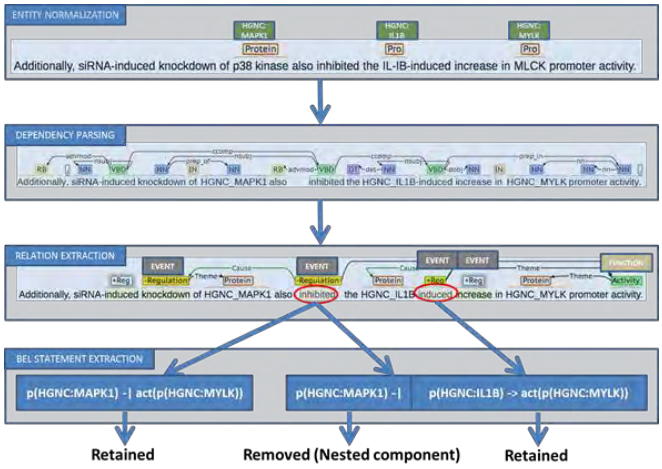


Fig. 2. Illustration of the function of individual components of BELMiner using an example sentence (PMCID: PMC3677168)

In the biomedical text, the presence of coordinated entities is quite common. For example, in the phrase "ERK1 and -2" while it is easily possible to detect ERK1 as gene the second entity ERK-2 is difficult to identify unless the entities are decomposed into two entities ERK1 and EKR-2. We integrated SimConcept [16] into the current BELMiner 2.0 pipeline in order to distinctly identify the composite entities and normalize them to the right database identifiers. Composite names and entity co-ordinations are common in bio-medical literature and their resolution has been shown to improve the performance of gene/disease detection and normalization by 1% [16].

C. *Improved syntactic and semantic features in BELMiner 2.0*

One of the shortcomings in the BELMiner was that its ability to identify long distance relations that occur beyond clausal boundaries are limited. In order to overcome this issue, we used the new Stanford Parser 3.8 [17] that identifies extended dependencies such as anaphora/co-reference resolution, appositives and dependencies that occur beyond clausal boundaries with greater accuracy. Subsequently, we used a list of trigger words to identify the functions and relations (increase or decrease) to identify them in the text.

Our approach to identifying appropriate arguments for function and relations involve a graph based traversal [18]. For example, consider an example sentence shown in Figure 2. The system identified *induced*, *knockdown*, *inhibited* and *increase* as the trigger words that describe bio-medical events. Similarly, the term *activity* was identified as a "function". From each of the event phrases, the relation extraction module traverses along the dependency graphs until it reaches an entity or a function word. During graph traversal, the system considers certain properties of the node such as the semantic type of the node and properties such as negations. It also identifies the successive double negation of events along the traversal path to correctly identify the type of the main event. Consider the phrase "siRNA-induced **knockdown** of ***p38 kinase*** also **inhibited**" in the example sentence shown in Figure 2. There is a possibility to mis-identify "inhibited" as a negative regulation by just considering that word alone. The event "inhibited" takes "knockdown" as the causal argument, which in turn takes entity "p38 kinase" as its argument. Since both events belong to the Negative_Regulation class, the double negation classifies the type of the main event "inhibited" as "Positive_Regulation" class. In this case, the system identifies the correct relation "p(HGNC:MAPK1) ->".

TABLE I. MAPPING NLP FUNCTION AND EVENT TYPES TO BEL FUNCTIONS

Keywords	Type	BEL functions
Physical interaction, binding, complex formation	Function	complex
Gene expression, Transcription	Function	rnaAbundance()
Translocation	Function	tloc()
Phosphorylation, acetylation, methylation	Function	pmod(P/A/M)
Degradation,	Function	deg()
Activity	Function	act()
increases, activation, induce	Event (Relation)	increases

inhibit, block	Event (Relation)	decreases
Directly increases, direct activation, directly induce	Event (Relation)	directlyIncreases
Directly inhibit, directly block	Event (Relation)	directlyDecreases

D. Mapping textual relations to BEL statements

We retained a simple rule-based approach to map the relation extraction output to a formal BEL statement that we have described in our earlier paper [5]. Table 1 lists the mapping between NLP functions and events into BEL functions. We finally filter out the incomplete BEL statements that do not fit the syntax of BEL formalism and recursive BEL statements, as they were not considered for the evaluation in this task.

III. RESULTS AND DISCUSSION

We used the training and test data for BioCreative V BEL track task 1 from 2015 [4] for fine tuning the performance of updated BELMiner. We ran the new BELMiner 2.0 on the test data provided as part of BioCreative BEL task 2017. Similar to the 2015 tasks the performance of systems was performed at different levels namely, Term-Level, Function-Level, Relationship-Level, Full Statement and Overall Evaluation. It was further carried out in two phases i) without named entities and after providing the gold standard entities. Table 2 outlines the results of the system for both runs of the system with and without the gold standard entities respectively. We submitted two runs of BELMiner output during Phase I and Phase II. The only difference between the two runs is that the first run consists of partial BEL statements where the missing entities (arguments) are replaced with PH:Placeholder. The standard metrics namely Precision, Recall, and F-measure was used to evaluate the performance of the system.

TABLE II. PERFORMANCE OF BELMINER ON BIOCREATIVE BEL TASK 2017 (WITH AND WITHOUT GOLD STANDARD ENTITIES)

Class		Entities from Gold standard			Entities from NER		
		Pre (%)	Rec (%)	F-Mes (%)	Pre (%)	Rec (%)	F-Mes (%)
Term (T)	R1	90.56	**84.92**	**87.65**	78.18	**70.49**	**74.14**
	R2	**90.94**	82.3	86.40	**78.71**	67.87	72.89
Function Secondary (FS)	R1	79.07	**45.33**	57.63	70.83	**45.33**	**55.28**
	R2	**89.19**	44.0	**58.93**	**79.49**	41.33	54.39
Function	R1	77.08	**38.95**	51.75	56.6	**31.58**	**40.54**
	R2	**85.71**	37.89	**52.55**	**63.64**	29.47	40.29
Relation-Secondary (RS)	R1	95.75	**88.65**	**92.06**	89.62	**82.97**	86.17
	R2	**97.55**	86.99	91.92	**93.00**	81.22	**86.71**
Relation	R1	**76.70**	**59.21**	**66.83**	51.81	**37.72**	43.65
	R2	**76.70**	**59.21**	**66.83**	**52.12**	**37.72**	**43.77**
BEL Statement	R1	63.01	**40.35**	49.2	40.67	**26.75**	32.28
	R2	**64.34**	**40.35**	**49.6**	**41.22**	**26.75**	**32.45**
Pre – Precision; Rec – Recall; F-Mes – F-Measure R1 – Run1; R2 – Run2; R3 – Run3							

A. Term level performance

Figure 3 and 4 outlines the performance of the new BELMiner during Phase I and Phase II, respectively. Comparison between the first and second phase between the best systems reveal a 12% difference between Phase I and Phase II. The native BELMiner identified 275 entities in total out of which only 215 were found to be correct when compared against the 305 gold standard entity annotations. With gold standard entities, the system extracted 259 entities correctly. Out of the 305 gold standard annotations, there were 247 protein, 36 chemical, 20 biological process, and 2 diseases annotations. During Phase I the best system extracted 169 protein annotations (F-mes: 72%), while we observed a steep increase during Phase II. We did not observe much of a difference between two phases for chemical entity detection. We saw only 0.5% increase in the overall F-measure for chemical detection. There is a substantial increase in the biological process performance (11% increase) between Phase I and Phase II. The total number of protein annotations was more than the other two and they might have reflected in the overall numbers.

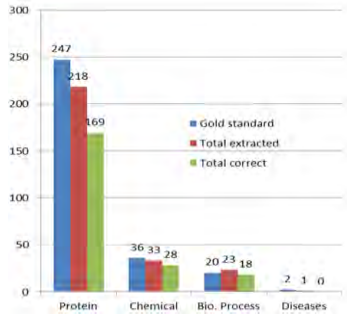


Fig. 3. Term (category wise) level evaluation during Phase I.

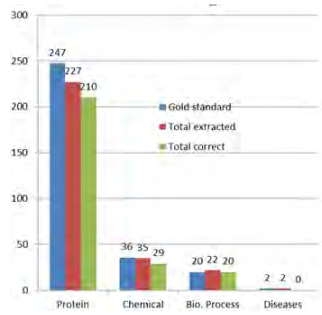


Fig. 4. Term (category wise) level evaluation during Phase II.

B. Performance of BELMiner in identifying functions

The ability of the BELMiner in identifying the functions (secondary functions) seems to be its key bottleneck. There were 75 functions annotated in the gold standard out of which the system identified only 34 correctly. There were totally 6 functional categories namely, activity, complex formation, secretion, protein modification, degradation, and translocation. The performance seems to be the lowest among the activity (F-Mes: 24%). The system really performs well in identifying the post-translational modification (F-mes: 92.31%). We haven't completed our error analysis to understand how to improve the system better in its ability to identify the functions in the text. Figure 5 outlines the performance of BELMiner on secondary functions belonging to different categories.

C. Performance of BELMiner in extracting the relations

The difference in the performance of the system in extracting the relations during Phase I and Phase II is very huge. We observed a huge increase in 23% overall F-measure when we used the gold standard entities for named entities. The increase in the NER performance directly

correlates with the relation extraction. However, the evaluation at secondary functions level revealed that we did not observe a huge difference (increase in 5%) between the two phases.

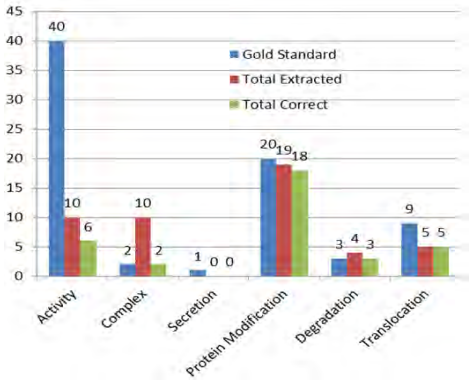


Fig. 5. Secondary function evaluation

D. *Performance of BELMiner in extracting the overall BEL statements*

The trend in complete BEL statement extraction is similar to that of relation extraction between the two phases. We observed nearly a 17% increase in the performance of BEL statement extraction between the two phases. Compared to the performance of BEL statement extraction in 2015, we observed nearly a 15% overall improvement in the performance of BEL statement extraction.

IV. CONCLUSIONS

In this work, we described a generic graph-based traversal on the dependency graphs constructed out of entity normalized sentences. We discussed the overall impact of different changes to the BELMiner on different tasks in BEL statement extraction on BioCreative BEL 2017 extraction task. We observed significant improvement in the state of the art in BEL statement extraction. Similar to the experience we had last time the performance of term level extraction played a major role in the overall improvement of BEL statement extraction. We believe that we can further improve the performance of the system through systematic analysis of the errors.

ACKNOWLEDGMENT

We acknowledge funding from National Institutes of Health (NIH) , grant number NIH-NCATS: 1OT3TR002019-01, "Biomedical Data Translator Technical Feasibility Assessment and Architecture Design", and Mayo clinic Research Foundation that supported this work.

REFERENCES

1. Wei, C.H., Kao, H.Y. and Lu, Z., *PubTator: a web-based text mining tool for assisting biocuration.* Nucleic acids research, 2013: p. gkt441.
2. Madan, S., Hodapp, S., Senger, P., Ansari, S., Szostak, J., Hoeng, J., Peitsch, M. and Fluck, J., *The BEL information extraction workflow (BELIEF): evaluation in the BioCreative V BEL and IAT track.* Database, 2016(Jan 1).
3. Fluck, J., Madan, S., Ellendorff, T.R., Mevissen, T., Clematide, S., van der Lek, A. and Rinaldi, F., *Track 4 Overview: Extraction of Causal Network Information in Biological Expression Language (BEL).* 2015: p. 333-346.
4. Fluck, J., Madan, S. and Ansari, S., *Training corpora for the extraction of causal relationships coded in Biological Expression Language (BEL).* . Database, 2016(This issue).
5. Ravikumar, K.E., Rastegar-Mojarad, M., and Liu, H., *BELMiner: adapting a rule-based relation extraction system to extract biological expression language statements from bio-medical literature evidence sentences.* . Database, 2017. **1**(baw156).
6. Nunes, T., et al., *BeCAS: biomedical concept recognition services and visualization.* Bioinformatics, 2013. **29**(15): p. 1915-1916.
7. Torii, M., Hu, Z., Wu, C.H. and Liu, H., *BioTagger-GM: a gene/protein name recognition system.* Journal of the American Medical Informatics Association, 2009. **16**(2): p. 247-255.
8. Leaman, R., and Lu, Z., *TaggerOne: joint named entity recognition and normalization with semi-Markov Models.* . Bioinformatics, 2016. **32**(18): p. 2839-2846.
9. Schwartz, A.S.a.H., M. A. *A simple algorithm for identifying abbreviation definitions in biomedical text*. in *Pacific Symposium on Biocomputing*. 2003. Hawaii.
10. Morgan, A.A., Lu, Z., Wang, X., Cohen, A.M., Fluck, J., Ruch, P., Divoli, A., Fundel, K., Leaman, R., Hakenberg, J. and Sun, C., *Overview of BioCreative II gene normalization.* Genome biology, 2008. **9**(Suppl 2): p. S3.
11. Apweiler, R., Bairoch, A., Wu, C.H., Barker, W.C., Boeckmann, B., Ferro, S., Gasteiger, E., Huang, H., Lopez, R., Magrane, M. and Martin, M.J., *UniProt: the universal protein knowledgebase.* . Nucleic acids research, 2004. **32**(suppl 1): p. D115-D119.
12. Maglott, D., Ostell, J., Pruitt, K.D. and Tatusova, T., *Entrez Gene: gene-centered information at NCBI.* . Nucleic acids research, 2005. **33**(suppl 1): p. D54-D58.
13. Povey, S., Lovering, R., Bruford, E., Wright, M., Lush, M., and Wain, H., *The HUGO gene nomenclature committee (HGNC).* . Human genetics, 2001. **109**(6): p. 678-680.
14. Bult, C.J., Eppig, J. T., Kadin, J. A., Richardson, J. E., Blake, J. A., and Mouse Genome Database Group, *The Mouse Genome Database (MGD): mouse biology and model systems.* Nucleic acids research, 2008. **36**(suppl_1): p. D724-D728.
15. Hahn, M.A.V.E.G., and Surdeanu, P. T. H. M. *A domain-independent rule-based framework for event extraction.* . in *ACL-IJCNLP*. 2015.
16. Wei, C.H., Leaman, R., and Lu, Z. *SimConcept: a hybrid approach for simplifying composite named entities in biomedicine*. in *5th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics*. 2014.
17. Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J. R., Bethard, S., and McClosky, D. *The stanford corenlp natural language processing toolkit*. in *ACL (System Demonstrations)*. 2014.
18. Ravikumar, K.E., Wagholikar, K. B., Li, D., Kocher, J. P., and Liu, H., *Text mining facilitates database curation-extraction of mutation-disease associations from Bio-medical literature.* . BMC bioinformatics, 2015. **16**(1): p. 185.

Generating Biological Expression Language Statements with Pipeline Approach and Different Parsers

Po-Ting Lai[1], Ming-Siang Huang[2], Wen-Lian Hsu[1], Richard Tzong-Han Tsai[3,*]
[1] Department of Computer Science, National Tsing-Hua University, Taiwan, R.O.C.
[2] Bioinformatics Program, Taiwan International Graduate Program, Institute of Information Science, Academia Sinica, Taipei, Taiwan, R.O.C.
[3] Department of Computer Science and Information Engineering, National Central University, Taiwan, R.O.C.

***Abstract*—In this paper, we describe our approach for the task 1 of the BioCreative VI Biological Expression Language (BEL) track. Our pipeline system is based on the BEL statement generation system, BelSmile. For the BioCreative VI BEL task, many components of BelSmile were updated. 1) We replaced the Conditional Random Fields (CRFs)-based approach with statistical principle-based approach (SPBA) for gene mention recognition. 2) A new verbal patterns were developed for function classification. 3) To improve SRL, we ensemble different semantic role labeling (SRL) parsers. 4) Our system is able to generate BEL statement even the relation is not described in a subject-verb-object (SVO) format. In the task 1, our best configurations achieved an F-score of 22.99% on the stage 1, and an F-score of 31.79% on the stage 2.**

Keywords—Biological Expression Language; Semantic Role Labeling; Named Entity Recognition; Relation Extraction

I. INTRODUCTION

The goal of the BioCreative VI Biological Expression Language (BEL) task 1 is that given a biological evidence sentence, the participants have to generate its corresponding BEL statement(s). The main challenges of this task are as following. First, the task contains many stages including named entity recognition (NER), named entity normalization (NEN), function classification, and relation classification. Therefore, developing a BEL statement generation system is more complicated than developing a single component. Second, the positions of named entity (NE), function and relation keyword are not provided in the training set. Therefore, the training set cannot be used to tune machine-learning models without appropriate preprocessing.

To tackle above challenges, we developed our system based on our previous pipeline system, BelSmile (1). Table I summarizes the resources and methods used for this task, and the differences with our previous system. The one with '*' symbol means that it was used for this task but not used in BelSmile.

BelSmile and our new system for BioCreative VI BEL task 1 differs in several ways. First, the CRFs-based gene mention recognition component was replaced by SPBA NER (2). Second, the verbal patterns were developed for function classification component. Third, for semantic role labeling, we ensemble our SRL parser, RCBiosmile (3), and a commonly-used parser, Enju (4). Lastly, our system could generate BEL statement even when the relation was presented in temporal and location statement.

This paper is arranged as follows. In Section II, we introduce our system for the BioCreative VI BEL track task 1. The configurations of each submission are described in Section III. In Section IV, we discuss the advantage of our approach. Section V concludes the paper and gives a future work.

TABLE I. THE RESOURCES AND METHOD USED FOR THIS TASK

Component	Method	Training set	Dictionary
Biological Process Recognition	Dictionary		BEL dictionary
Chemical Recognition	CRFs + dictionary	***BioCreative V CEMP**	ChEBI
Disease Recognition	Dictionary		BEL dictionary
Protein Recognition	***SPBA** + dictionary	JNLPBA + ***BioCreative VI CPRO**	Entrez
Function classification	Non-verbal pattern + ***verbal pattern**		
Semantic Role Labeling	RCBiosmile + ***Enju**	BioProp	
Relation Extraction	SRL + ***time/location rules**		

II. SYSTEM DESCRIPTION

In this section, we first introduce the named entity recognition components used in our system, and then introduce our normalization component. Third, we introduce our pattern-based approach for function classification. Then we introduce two different semantic role labeling components used for relation classification, and how we combine them. Lastly, we introduce how to integrate all components for BEL statement generation.

A. Named Entity Recognition

Here we briefly describe our named entity recognition components, which were used in our previous works (1, 2, 5, 6).

Chemical Recognition: We used NERChem (5) for chemical recognition. It was a CRF-based system which used the two-stage tokenization, consisting of GENIA tokenization and symbol tokenization. It used SOBIE (Singleton-, Outside-, Beginning-, Inner- and Ended-Named Entity) tag scheme and empirical feature set, including word, POS, affix, orthographical, word shape, syntax and NE features. We trained NERChem on the BioCreative IV chemical corpus to recognize chemicals.

Disease and Biological Process Recognition: For diseases and biological process recognition, the dictionaries provided by the BEL task were used to develop our dictionary-based recognition component. We used the longest matching algorithm to recognize both disease and biological process.

Gene Mention Recognition: We used SPBA (6) for gene mention recognition. It integrated the advantages of robustness of machine learning model and interpretability of pattern-based approach. SPBA was developed based on our revised version of JNLPBA corpus and BioCreative V.5 Gene and Protein Related Object Recognition (GPRO) corpus (2, 7).

B. Named Entity Normalization

Since named entity (NE) may not exactly match its corresponding dictionary names, the normalization process used heuristic rules to expand the query of a NE. The heuristic normalization rules, such as converting to lowercase and removing symbols and the suffix *'s'*, used in our previous works (1) are employed to expand both NE and dictionary names. Moreover, additional rules were developed to normalize proteins. For example, if a protein complex *"A/B complex"* is failed to be mapped into identifier, we will try to normalize it by separating it into two proteins. Table II shows the resources used in normalization.

TABLE II. THE RESOURCES USED FOR RECOGNITION AND NORMALIZATION

Type	Dictionary
Chemical	ChEBI
Gene mention	Entrez gene (human and mouse)
Biological Process	BEL dictionary
Disease	BEL task

C. Function Classification

Function classification component classifies the molecular activity of the NEs into transcription and phosphorylation activity etc. We used a pattern-based approach to classify the functions of the NEs. Our patterns are divided into two categories: non-verbal and verbal patterns.

In non-verbal pattern, a pattern consists of NE(s) and molecular activity keyword(s). Table III shows some examples of our non-verbal patterns, which were written by our domain experts.

In verbal pattern, each pattern consists of predicate and arguments. In the next section, we will introduced how to generate the predicate-argument-structure (PAS) of a given sentence. Table IV shows some examples of the verbal patterns, and how to transform the PAS into the BEL function statement.

TABLE III. EXAMPLES OF NON-VERBAL FUNCTION PATTERNS

Function	Example Pattern
molecularActivity (act)	<Protein/> activity
complexAboundance (complex)	<Protein/>/<Protein/> complex
degradation (deg)	<Protein/> degradation
proteinModification (pmod)	phosphorylation of <Protein/>
translocation (tloc)	translocation of <Protein/>

TABLE IV. EXAMPLES OF VERBAL FUNCTION PATTERNS

Function	Example Pattern	BEL statement
act	<AgentNE/> <Verb>activates</Verb> <PatientNE/>	<AgentNE/> increase act(<PatientNE/>)
complex	<ProteinA/> is complexed with <ProteinB/>	complex(<ProteinA/>,<ProteinB/>)
pmod	<AgentNE/> <Verb>phosphorylates</Verb> <PatientNE/>	<AgentNE/> increase pmod(<PatientNE/>, "P")

D. Semantic Role Labeling

Before we introduce our relation extraction component, we briefly introduce two SRL components, RCBiosmile and Enju, used in our relation classification component.

RCBiosmile: RCBiosmile (3) is a Markov-Logic-Network (MLN)-based SRL labeler that employs patterns to select candidate semantic roles for each argument and uses MLN (8) to learn and predict the semantic role of each argument. It was trained on BioProp (9).

Enju: Enju is a commonly-used semantic parser (4), which can extract agent and patient arguments. We selected it, since we found that it seems more accurate than other open-source non-biomedical domain SRL parsers in the BEL training set.

E. Combining Different SRL Systems

Here, we describe the procedure that used to combine SRL parsers as follows.

Step l: Both Enju and RCBiosmile were used to parse the predicate-argument-structure (PAS) of a given sentence.

Step 2: The agent argument, patient argument and predicate of the Enju PAS were collected into a list.

Step 3: All arguments and predicate of the RCBiosmile PAS were collected into a list.

Step 4: In the Enju, the arguments like negation, time and location were not labeled as negation, temporal and location arguments. Therefore, if Enju predicts the same predicate with

Identify applicable sponsor/s here. If no sponsors, delete this text box (*sponsors*).

RCBiosmile. For these arguments, we will add these arguments generated by the RCBiosmile into the Enju's argument list.

Step 5: According to our observation on the BEL training set. We found that the RCBiosmile is more accurate than Enju while the predicted predicate is defined in the BioProp corpus. E.g. the predicate *"phosphorylate"*. However, these predicates often appear in the BEL training set. Therefore, we combined the results of two parsers by using predicate. For the predicates which were used in BioProp, we will use the arguments generated by RCBiosmile, and for the rest predicates we will use the arguments generated by Enju.

F. BEL Statement Generation

In this section, we describe how to extract the cause-theme-event relationship from PASs and transform them into BEL relations.

Given a sentence, combined SRL parser is used to parse it, and we will retrieve one or more PAS(s). Each PAS contains the arguments corresponding to the predicate. To extract cause-theme-event relationship, we map the predicate into the verb; map the abundances/processes which are inside the boundaries of the agent argument and the patient argument into the cause and the theme respectively.

However some cause-theme-event relationships are not presented in subject-verb-object format. For example, given a sentence,

"Furthermore, the expression of Bach 2, which can form a heterodimer with mafG protein, was found to be greatly reduced, while Notch 1 expression was increased in mafG-deficient mice." --- *PMID:20813153*

In SRL, the phrase *"in mafG-deficient mice"* is labeled as a location argument of the predicate *"increased"*;*"Notch 1 expression"* is labeled as the patient argument of the predicate *"increased"*. Although, *"mafG-deficient mice"* and *"Notch 1 expression"* are not an agent-patient pair, but the sentence means that *"mafG-deficient mice"* would increase *"Notch 1 expression"*. Therefore, the BEL statement

"p(EGID:4097) -| p(EGID:4851)" should be generated.

"p(EGID:4097)" refers to *"mafG"*; "p(EGID:4851)" refers to *"Notch 1"*.

To solve such problem, we map the predicate into the verb; map the abundances/processes which are inside the boundaries of the agent/patient argument into theme; map the abundances/processes which are inside the boundaries of the location or temporal arguments into cause.

The BEL relationship type is then determined by the regulation keywords collected from the BioNLP corpora (10) where "Regulation" and "Positive_regulation" types are mapped into the *increases*, and "Negative_regulation" is mapped into the *decreases*. If the surrounding context of NE abundances has certain keywords, like *"inhibition"* and *"knockout"*, then we will reverse the relation type.

III. EXPERIMENT RESULTS

We participated in both stage 1 and 2 of the BioCreative VI BEL track task 1, and three runs were submitted for each stage. The configurations of all runs are as follows:

Stage1:

- Run 1 (our best): it used the combined SRL parser.
- Run 2: it only used Enju for SRL.
- Run 3: it only used RCBiosmile for SRL.

Stage 2:

- Run 1: it used the combined SRL parser.
- Run 2: it used RCBiosmile parser as baseline, and added Run 1 statements. Furthermore, if some sentences only generated function statements, we will also output them.
- Run 3 (our best): it used the Run1 as baseline, and added RCBiosmile-based statement. Furthermore, if some sentences only generated function statements, we will also output them.

Table V and VI show the different level performances of our runs. In stage 1, the Run 1 used the combined SRL parser achieved our best performance on the statement evaluation metric. In stage 2, the Run 3 achieved our highest performance.

TABLE V. THE PERFORMANCES OF STAGE 1

Class		Recall	Precision	F-score
Term	run1	50.49	84.62	**63.24**
	run2	44.59	81.93	57.75
	run3	46.89	88.27	61.24
Function	run1	27.37	44.83	**33.99**
	run2	24.21	43.4	31.08
	run3	25.26	47.06	32.88
Relation	run1	31.58	55.38	**40.22**
	run2	28.07	53.33	36.78
	run3	28.07	56.14	37.43
Statement	run1	17.54	33.33	**22.99**
	run2	15.35	31.82	20.71
	run3	15.35	33.98	21.15

TABLE VI. THE PERFORMANCES OF STAGE 2

Class		Recall	Precision	F-score
Term	run1	72.79	99.11	83.93
	run2	76.07	99.15	**86.09**
	run3	75.08	99.13	85.45
Function	run1	29.47	47.46	36.36
	run2	33.68	50.79	**40.51**
	run3	32.63	49.21	39.24
Relation	run1	46.93	73.29	57.22
	run2	46.49	70.67	56.08
	run3	47.37	73.47	**57.6**
Statement	run1	23.68	46.15	31.3
	run2	23.68	44.63	30.95
	run3	24.12	46.61	**31.79**

IV. Discussion

In this section, we take a sentence as an example to illustrate the advantage of using combined SRL parser .

"Pulse-chase biosynthetic labeling studies showed that AtT-20 cells expressed much less RESP18 than the endogenous prohormone, POMC, but that glucocorticoid treatment lowered POMC and raised RESP18 biosynthetic rates so that they were nearly equimolar." --- PMID:7988462

Two gold BEL statements should be generated.

"a(CHEBI:glucocorticoid) -| p(EGID:5443);"

"a(CHEBI:glucocorticoid) -> p(EGID:389075)"

where *"POMC"* and "*RESP18*" are "p(EGID:5443)" and "p(EGID:389075)" respectively; *"lowered"* and *"raised"* are "-|" and "->" respectively. RCBiosmile failed to generate the BEL statements, since both *"lowered" and "raised"* were tagged as "JJ", and thus caused incorrect SRL results. However, in Enju, both *"lowered" and "raised"* can be tagged as "VBD", and then generated correct agent and patient arguments for both *"lowered" and "raised"*. In contrast, sometimes Enju might generate incorrect SRL results, but RCBiosmile didn't. Therefore, to solve this problem, combined SRL parser could reduce such errors coming from individual SRL components.

V. Conclusion and Future Work

In the future, we would like to apply the SPBA to tackle other NE types like chemical, disease and biological process. Using multiple components, the system performs better than using single component. Therefore, we would like to integrate different state-of-the-art systems in the future. The results of the BEL statement generation are depended on individual components. We also like to use deep learning-based approaches to enhance individual components like semantic role labeling.

References

1. Lai, P.-T., et al., BelSmile: a biomedical semantic role labeling approach for extracting biological expression language from text. *Database*, 2016. p. baw064.

2. Lai, P.-T., et al. Statistical Principle-based Approach for Gene and Protein Related Object Recognition. in *Proceedings of the BioCreative V.5 Challenge Evaluation Workshop*. 2017.

3. Tsai, R.T.-H. and P.-T. Lai, A resource-saving collective approach to biomedical semantic role labeling. *BMC bioinformatics*, 2014. 15(1): p. 160.

4. Matsuzaki, T., Y. Miyao, and J.i. Tsujii, Efficient HPSG parsing with supertagging and CFG-filtering, in *Proceedings of the 20th international joint conference on Artifical intelligence*, 2007, Morgan Kaufmann Publishers Inc.: Hyderabad, India. p. 1671-1676.

5. Tsai, R.T.-H., Y.-C. Hsiao, and P.-T. Lai, NERChem: adapting NERBio to chemical patents via full-token features and named entity feature with chemical sub-class composition. *Database: The Journal of Biological Databases and Curation*, 2016 p. baw135.

6. Tsai, R.T.-H., et al., NERBio: using selected word conjunctions, term normalization, and global patterns to improve biomedical named entity recognition. BMC bioinformatics, 2006. 7(Suppl 5): p. S11-S11.

7. Pérez-Pérez, M., et al. Evaluation of chemical and gene/protein entity recognition systems at BioCreative V.5: the CEMP and GPRO patents tracks. in *Proceedings of the BioCreative V.5 Challenge Evaluation Workshop*, 2017.

8. Richardson, M. and P. Domingos, Markov logic networks. *Machine Learning*, 2006. 62(1): p. 107-136.

9. Tsai, R., et al., Semi-automatic conversion of BioProp semantic annotation to PASBio annotation. *BMC bioinformatics*, 2008. 9(Suppl 12): p. S18.

10. Kim, J.-D., et al., Overview of BioNLP Shared Task 2011, in *Proceedings of the BioNLP Shared Task 2011 Workshop*, 2011, Association for Computational Linguistics: Portland, Oregon. p. 1-6.

Automatic Extraction of BEL-Statements based on Neural Networks

Mehdi Ali[1,2], Sumit Madan[2], Asja Fischer[1], Henning Petzka[1] and Juliane Fluck[2]
[1]University of Bonn, 53012 Bonn,
[2]Fraunhofer Institute for Algorithms and Scientific Computing, Schloss Birlinghoven, 53754 Sankt Augustin

***Abstract*— The automatic extraction of biomedical relations and entities from text has become extremely important in systems biology. For coding the extracted information, the Biological Expression Language (BEL) can be used. A BEL-statement consists of a subject (entity), a predicate (type of relationship), and an object (entity or a further BEL-statement). This paper describes a system based on neural networks (NNs) to extract BEL-statements in the context of the BioCreAtivE 2017 track 3 (task 1) challenge. In our approach, the overall problem is divided into four subtasks: (i) the detection of named entities (NER), (ii) deciding whether a pair of entities participate in a relation, (iii) determining which of the entities participating in a relation is the subject/object entity, and (iv) extracting the type of the relation. By merging the solutions of the subtasks, the BEL-statements are generated. Except for the named entity recognition, (convolutional) NNs were used to solve the tasks. The results show that a neural net based approach is reasonable to use for the extraction of biomedical relations. The limitations of our system are related to the small size (compared to other NN-based applications) of the data set. We argue that by overcoming this limitation, promising results can be expected from NN-based approaches in future.**

Keywords*— *Text Mining; Convolutional Neural Network; Relation Extraction; Biological Expression Language (BEL)

I. Introduction

To be able to analyze the immense amount of biological data, automated systems are required. In order to perform an automated analysis, the knowledge must be available in a computational representation. Unfortunately, a big amount of the existing knowledge is only available in form of unstructured scientific texts. Therefore, it is of interest to develop systems that extract information from scientific publications and represent the information in a machine interpretable form. The Biological Expression Language (BEL)[1] is a domain-specific language, which makes it possible to capture the extracted knowledge in a structured representation (1). Relationships encoded in BEL are triples consisting of a subject, a predicate, and an object. For example, the BEL statement "p(HGNC:TLR2) increases cat(p(HGNC:CASP1))" can be extracted from the sentence "Interestingly, BLP also activates caspase 1 through TLR2, resulting in proteolysis and secretion of mature IL-1beta" (PMID:10880445) (2).

The BioCreative VI BEL track task 1[2] offers a platform to compare systems for extracting biological knowledge out of free-texts and saving the gathered information as BEL-statements (3). This task is carried out in two stages: i) without information about named entities and ii) with gold standard named entities. The evaluation scheme designed by the authors consider several structural levels of a BEL statement such as term (for e.g. HGNC:CASP1), function (for e.g. catalytic or kinase activity), and relation (for e.g. increases or decreases) level. For the latter two levels, two additional secondary levels are available. Due to time constraints, we did not participate in the function prediction.

In the last years, neural network (NN) based approaches have become popular in Natural Language Processing (NLP) (4–7). Especially convolutional neural networks (CNNs) and recurrent neural networks (RNNs) are used in text mining. The advantage of neural networks is that they are capable of learning features by themselves (representation learning) by comprising features learned in the different layers of the network in a hierarchical manner. In the low-level layers, basic features are learned and in the higher layers, more complex features are extracted based on the low-level features. This allows to overcome the process of feature engineering or at least to reduce this process (8).

Quan et al. (5) and Hua et al. (9) show that neural networks can successfully be used to extract biomedical relations from scientific publications. Quan et al. use a multichannel CNN approach to extract protein-protein (PPIs) and drug-drug interactions (DDIs). Their system is able to predict associations between pairs of entities without determining which of the entity in an association is the subject or object and without giving details about the type of the relation. For the BioCreative VI BEL task 1, we used this work as a foundation to develop our NN-based system that is capable of extracting BEL-statements out of sentences without involving the complex and time-consuming process of manual feature engineering.

In the following we give a short description of our system in Section II, present the achieved results in Section III, and discuss the results and provide a future outlook in Section IV.

[1] http://openbel.org

[2] http://www.biocreative.org/tasks/biocreative-vi/track-3/

II. System Description

Our NN-based system has a component based architecture in which each component has a specific task. Since the components can easily be exchanged, the system remains flexible. To extract BEL-statements out of the sentences four subtasks must be solved. The first sub-task (and the only one not solved by an NN in our setting) is entity recognition and normalization. The remaining tasks are formulated as binary classification problems (see Fig. 1) and each one is solved by an NN. For each pair of entities, the system has to predict, whether the sentence describes a relation between them. Then, for the case a relation exists, it must determine which of the entities corresponds to the subject and which to the object, respectively. The last step is to extract the type of the relation ("increases" or "decreases"). Based on the predictions for the four sub-tasks, a BEL-Statement is created for a pair of entities, if the existence of a relation is predicted. We use a multi-channel CNN architecture as proposed by Quan et al. (5) in all NN-based sub-tasks. For each task, we train a separate model. In the following, the four single steps and the applied CNN model are explained in more detail.

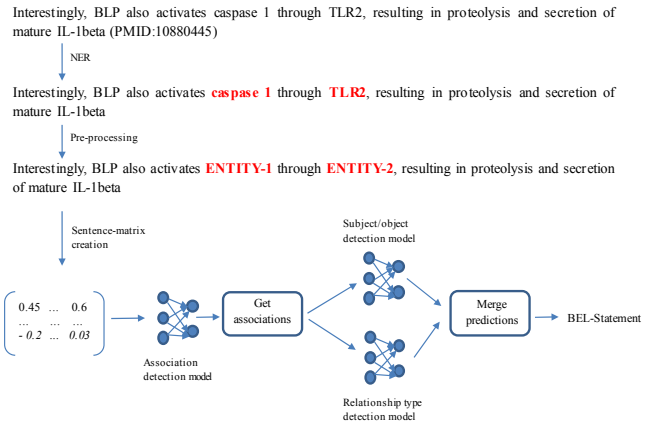


Fig. 1. BEL-Statements extraction workflow

A. Named Entity Recognition (NER)

The first step is the extraction of named entities from the sentences. For this task, the rule and dictionary based software ProMiner (10) is used. It contains several terminologies to detect named entities. We use ProMiner with the terminologies listed in Table I. On the training set we used ProMiner to find the offsets of the annotated entities. For the test set we consider all detected entities for further predictions.

B. Association Detection Model

The second sub-task is to decide, whether a sentence describes an association between two entities or not (independently from any information about subject/object and the relationship type). Since we use a supervised machine learning approach, we need training instances from both classes to train our models. In the training set only positive examples (without any position information of the entities in question within the sentence) are annotated, meaning that only examples of related entities are given. If a sentence does not mention a relation between two occurring entities, this fact is not captured. However, to train a binary classifier, also negative examples are required. Therefore, we create artificial negative examples based on the assumption that, if for a pair of entities no relation is annotated in the training set, the sentence doesn't describe a relation between these entities. This strategy can produce false positives in the sense that a negatively annotated relation might in fact hold because it is not guaranteed that the curator annotated all relations in a sentence (11). The advantage of this approach is that, given a reliable NER tool, the automated creation of negative examples can be realised with very low effort. Our model was trained based on 6,389 instances comprised of 4,633 positive and 1,756 negative examples. To evaluate our model, we applied a 10-fold cross-validation. The results are depicted in Table II. Although the data set is imbalanced the results are quiet promising, especially for the "Association"- class.

TABLE I. Terminologies used for NER

Entity Class	*Resources*	*OpenBEL Namespace*
Human genes/proteins	EntrezGene/Uniprot	HGNC
Mouse genes/proteins	EntrezGene/Uniprot	MGI
Chemicals	ChEBI	CHEBI
Diseases	MeSH disease subtree	MESHD
Biological processes	Gene Ontology biological processes	GOBP

TABLE II. 10-fold cross-validation results for association detection model

Class	*Recall*	*Precision*	*F1-Score*
Association	96.5%	90.7%	93.4 %
No-Association	73.9%	88.9%	80.7%

C. Subject/Object Detection Model

The third sub-task is to decide which of the entities in an entity pair is the subject and which the object. To train this model, only positive examples are needed. Since the subject and object information is contained in the training set, a training instance can be directly created as follows. For each relation, the subject and the object are extracted. If the subject occurs before the object in the sentence, the instance is assigned to the class "Subject First" otherwise to the class "Object First". This training set consists of 4,633 examples from which 3,156 instances belong to the class "Subject First" and 1,477 instances to the class "Object First". This data set is imbalanced, too, and the results for the 10-fold cross-validation (see Table III) show that there is still room for improvement.

TABLE III. 10-fold cross-validation results for subject/object detection-model

Class	*Recall*	*Precision*	*F1-Score*
Subject First	88.5%	81.5%	84.9%
Object First	56.9%	70.3%	62.3%

D. Relationship Type Detection Model

The last subtask is to determine the type of the relationship of an entity pair participating in an association. As for the subject/object detection subtask no artifical negative instances have to be created. The training set contains the relationship types "increases", "directly increases", "decreases" and "directly decreases", but for the BioCreative VI BEL task 1 "directly increases" is mapped to "increases" and "directly decreases" is mapped to "decreases", so that there are only two types of relationships to predict. The neural network for this task is trained based on 4,325 instances consisting of 3,103 "increases" and 1,222 "decreases" examples. The results are shown in Table IV.

TABLE IV. 10-fold cross-validation results for relationship type detection model

Class	*Recall*	*Precision*	*F1-Score*
Increases	91.8%	80.8%	85.7%
Decreases	39.5%	66.4%	48.2%

The model lacks in the correct identification of the class "decreases", especially the recall is low. We argue that this is due to the unbalanced data set. The class "increases" contains 71.7% of the instances while only 28.3% of the instances belong to the class "decreases". To solve this issue, we experimented with an oversampling strategy by simply copying the "decreases"-examples. However, no significant improvement could be achieved, but the time for training the model rose, so that we decided to train the final model with the initial data set.

E. Architecture of the Multichannel CNN

For all three models described in Section II. B-D multichannel CNNs are trained (see Fig. 2). The embedding layer contains the representations of the input sentence. The idea of the multichannel CNN is to use different input channels for different representations of the sentence. Different Word2Vec models (4) are used to transform each word of the sentence into a vector representation. Based on these word-vectors, a sentence-matrix is generated and passed to the network as input. In the sentence-matrix, each column represents a word and the number of rows indicate the dimension of each word-vector. For our networks, we use four Word2Vec models [3] trained by (12), which are based on PubMed, PubMed Central (PMC), and Wikipedia texts, respectively.

In the convolutional layer, local features are computed by applying a convolutional operation on each sentence-representation (see Fig. 2). For each sentence-representation we retrieve a scalar value every time the sliding window of the convolution operation is shifted. The scalar values of each shift are summed up and passed to the activation function. The rows of a new matrix represent the results of the convolutions created by different kernels/filters (in this case 4 filters). At the end of the convolutional layer a max-pooling operation creates a feature vector by extracting the most significant features, taking for each column the biggest value. The feature vector is passed to a fully-connected layer and the result of the fully-connected layer is given as input to a softmax classifier producing the predictions. In the convolutional layer and in the fully-connected layer the Exponential Linear Unit (ELU) is used as the non-linear activation function.

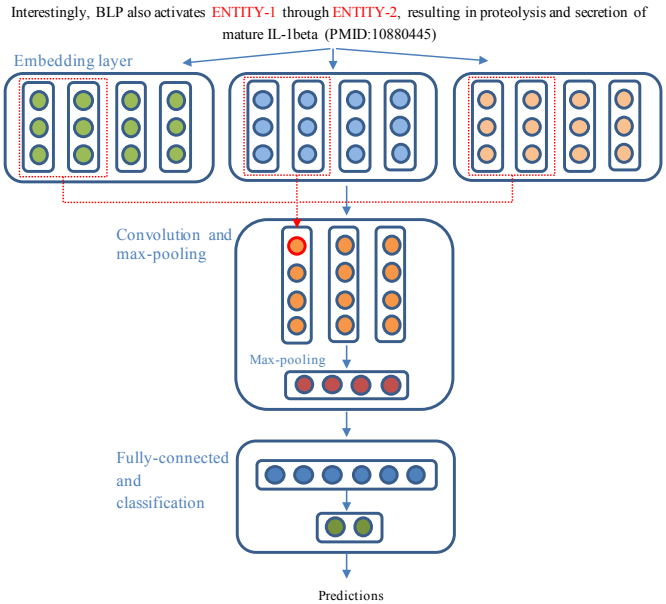


Fig. 2. Multichannel convolutional neural network architecture (5,9)

F. Hyper-Parameters of the Multichannel Convolutional Neural Network

The selection of appropriate hyper-parameters has a big influence on the performance of the model. We tested different values, but we did not apply any automated hyper-parameter optimization techniques. The selected set of parameters that is used for all three detection models (B-D) is depicted in Table V.

TABLE V. Hyper-Parameters of the trained models

Parameter	*Value*
Learning rate	0.01
Number of filters	200
Filter size	7
Dimension of word-vectors	200
Hidden neurons in the fully connected layer	500
Activation function in convolution layer	Exponential Linear Unit
Activation function in fully conn. layer	Exponential Linear Unit
Loss function	Cross-Entropy
Optimizer	Adagrad
Batch size	20
Training steps	100.000

[3] http://bio.nlplab.org/

III. Results on the Test Set 2017

For each stage of the BioCreative VI BEL task 1 challenge, we handed in three submissions. The difference between Submission 1 and 3 is that the models used in Submission 3 are trained on a reduced training set. For the models of Submission 1, all sentences of the training set with a length up to 75 tokens were considered whereas for the models in Submission 3 the maximum sentence length was chosen to be 65 tokens. We followed this strategy to handle two different distributions of sentence lengths. For Submission 2, we use the models of Submission 1 with the additional rule that whenever two entities predicted to be in a relation have a distance greater than 10 tokens, the prediction is omitted. We argue that an entity which occurs in one part of the sentence is likely not related to an entity which occurs in a distant position of the sentence, especially considering the fact that sentences can contain the same entities multiple times at different positions. The best results are reached with the first submission and its results are presented in Table VI for stage (i) and in Table VII for stage (ii). The most significant observation is the decrease of performance from the Relation-Secondary level to the Relation level, which again directly causes a decrease in the full statement level. Also not predicting any function of entities has a direct impact on the full statement level.

TABLE VI. Results of stage (i) on test set 2017

Evaluation-Level	*Recall*	*Precision*	*F1-Score*
Term	72.13%	81.18%	76.39%
Relation-Secondary	70.74%	60.45%	65.19%
Relation	35.96%	25.55%	29.87%
Full statement	20.61%	16.10%	18.08%

TABLE VII. Results of stage (ii) on test set 2017

Evaluation-Level	*Recall*	*Precision*	*F1-Score*
Term	84.60%	99.23%	91.33%
Relation-Secondary	83.00%	90.05%	86.36%
Relation	45.61%	41.60%	43.51%
Full statement	22.37%	25.00%	23.61%

IV. Conclusion and Future Work

We have presented the results of our participation in the BioCreative VI BEL track task 1. Our BEL extraction workflow is based on a multichannel CNN architecture motivated by Quan et al. (5). As features for our models, we used several pre-trained Word2Vec embeddings that were created on PubMed, PubMed Central, and Wikipedia texts. Our architecture reached for the relation level an F-score of 29.9% in stage (i) and 43.5% in stage (ii). The results indicate that for the extraction of BEL statements from natural language sentences a NN-based approach is reasonable.

The main difficulty when training NNs in this setting is the limited amount of training data. Especially for the relationship type detection model, we think that an increasing amount of training data can lead to a steep improvement of performance. Also the detection of functions of entities, which wasn't tackled in this work, should increase the performance on the full statement level.

Furthermore, it would be interesting to examine how the system performs with new, updated, and fine-tuned Word2Vec models. The models that were used during the training are already four years old. In particular the work of Bojanowski et al. (13) introducing an extended Word2Vec model seems promising for our use case. Besides, a hyper-parameter optimization approach will be integrated in our workflow, so that a better set of hyper-parameters can be detected. A further aspect to consider is the type of the neural network. Different architectures (e.g. recurrent neural nets) should be investigated.

We also plan to perform a detailed analysis of the predictions to detect the problems causing the performance decline. Finally, for use cases where limited training data is available a hybrid system containing machine learning components and rule-based modules should be taken into account.

Acknowledgement

We thank Linara Adilova and Sven Giesselbach for the fruitful discussions and their beneficial inputs, especially in generating and evaluating different Word2Vec models.

References

1. Slater, T. and Song, D. Saved by the BEL: ringing in a common language for the life sciences. 2012.
2. Aliprantis, A. O., Yang, R. B., Weiss, D. S., et al. (2000) The apoptotic signaling pathway activated by Toll-like receptor-2., *EMBO J.*, 19, 3325–36.
3. Rinaldi, F., Ellendorff, T. R., Madan, S., et al. (2016) BioCreative V track 4: a shared task for the extraction of causal network information using the Biological Expression Language, *Database*, 2016.
4. Mikolov, T., Chen, K., Corrado, G., et al. (2013) Distributed representations of words and hrases and their compositionality, *NIPS*, 1–9.
5. Quan, C., Hua, L., Sun, X., et al. (2016) Multichannel Convolutional Neural Network for Biological Relation Extraction, *Biomed Res. Int.*, 2016, 1–10.
6. Sutskever, I., Vinyals, O. and Le, Q. V (2014) Sequence to sequence learning with neural networks, *Adv. Neural Inf. Process. Syst.*, 3104–3112.
7. Zeng, D., Liu, K., Lai, S., et al. (2014) Relation Classification via Convolutional Deep Neural Network, *Coling*, 2335–2344.
8. Goodfellow, Ian, Bengio, Yoshua, Courville, A. Deep Learning http://www.deeplearningbook.org/.
9. Hua, L. and Quan, C. (2016) A Shortest Dependency Path Based Convolutional Neural Network for Protein-Protein Relation Extraction, *Biomed Res. Int.*, 2016, 1–9.
10. Hanisch, D., Fundel, K., Mevissen, H.-T., et al. (2005) ProMiner: rule-based protein and gene entity recognition., *BMC Bioinformatics*, 6 Suppl 1, S14.
11. Fluck, J., Madan, S., Ansari, S., et al. (2016) Training and evaluation corpora for the extraction of causal relationships encoded in biological expression language (BEL)., *Database (Oxford).*, 2016.
12. Pyysalo, S., Ginter, F., Moen, H., et al. (2013) Distributional semantics resources for biomedical text processing, *Proc. Lang. Biol. Med.*
13. Bojanowski, P., Grave, E., Joulin, A., et al. (2016) Enriching word vectors with subword information, *arXiv Prepr. arXiv1607.04606*.

Semantic Information Retrieval: Exploring Dependency and Word Embedding Features in Biomedical Information Retrieval

Majid Rastegar-Mojarad[1,2], Ravikumar Komandur Elayavilli[1], Yanshan Wang[1], Sijia Liu[1], Feichen Shen[1], Hongfang Liu[1]

[1]Department of Health Sciences Research, Mayo Clinic, Rochester, MN, USA
[2]University of Wisconsin-Milwaukee, Milwaukee, WI, USA

***Abstract*—We present an information retrieval system developed to retrieve evidence sentences for a given Biological Expression Language (BEL) statement. Previously, as a participant of the BEL challenge in BioCreative V, we proposed and developed a system called BELTracker, which mainly focused on lexical features. In the BEL challenge of BioCreative VI, we explored usage of syntactic and semantic features in identifying and ranking evidence sentences. Similar to BELTracker, the new system has 3 main components: indexing, retrieval, and ranking. In this system, we trained several classifiers for the ranking process. These classifiers have lexical, syntactic, and semantic features such as unigrams, bigrams, word embedding, and dependency-word embedding. Our evaluation showed that the new system obtained higher mean average precision on BioCreative V's test data when compared to BELTracker under full, relaxed, and context criteria. The challenge organizers provided 100 BEL statements as test data and we submitted 2 set of results 1) baseline (using ElasticSearch results), and 2) using the classifiers to re-rank the ElasticSearch results. The baseline system achieved 30.6% and 45.8% precision under full and partial criteria, respectively. Using various classifiers for ranking, the precision of the system increased to 31.6% and 50.2%, respectively, for full and partial criteria.**

Keywords—Biological Expression Language; Dependency-word embedding; Information Retrieval; Semantic Information Retrieval; Sentence retrieval; Word embedding

I. Introduction

The aim of information retrieval (IR) systems is to identify relevant resources to users' queries. Most IR systems consider scientific literature as resource and return abstracts or snippets of relevant literature as results. For some users, like physicians at point-of-care, reviewing or going through these results is time consuming and not practical. The ideal results for these users should be brief and pointed, especially if the users are looking for any evidence of relation between entities. For example, if a physician is interested in finding any evidence of interaction between 2 drugs, the evidence would most likely be in a single sentence (or 2 consecutive sentences). There have been several attempts in the biomedical domain to implement such IR systems which allow users to enter entities and type of relation between them and retrieve evidence sentences (1–6). The main limitation of these systems is focusing on unary or binary relations. They are unable to retrieve relevant evidence for biological observations which contain more than 2 entities and relations. The organizers of BioCreative V introduced a new track, called Biological Expression Language (BEL), which addresses this limitation. The track had 2 main tasks: 1) extracting BEL statements from a given sentence, and 2) identifying relevant sentences for a given BEL statement (7,8). BEL is 1 of the main representatives of biologic networks. A BEL statement has several components: normalized entities, functions, relations, namespace, and sequence positions. Table I illustrated 2 sentences and 3 extracted BEL statements.

TABLE I. EXAMPLES OF BEL STATEMENTS

Sentence	We showed that HSF1 is phosphorylated by the protein kinase RSK2 in vitro. We demonstrate that RSK2 slightly represses activation of HSF1 in vivo
Extracted BEL	**1:** kin (p (HGNC: RPS6KA3)) increases p (HGNC: HSF1, pmod (P)) **2:** kin (p (HGNC: RPS6KA3)) decreases tscript (p (HGNC: HSF1))
Sentence	Exposure of neutrophils to LPS or TNF-α resulted in increased levels of the transcriptionally active serine 133-phosphorylated form of CREB
Extracted BEL	p (MGI: TNF) increases p (MGI: CREB1, pmod (P, S, 133))
BEL Elements: Relationship, Function, Entity, Namespace, Sequence position	

BioCreative VI includes BEL track as well, and our team participates in the track and tries to improve our previous system, called BELTracker (9), and this paper describes our new system. One of the limitations of BELTracker is relying upon lexical features and heuristic approaches to rank returned results. In the new system, we try to address these issues by training several classifiers for ranking process and using semantic features along the way. Similar to BELTracker, the system has 3 main components: indexing, retrieval, and ranking components. First, we index all informative sentences in MEDLINE abstracts. For a given BEL statement, the

system constructs a query using all the elements in the BEL statement and retrieves the most relevant sentence from the index based on occurrence of the elements. Finally, using lexical and semantic features, the system ranks the results and returns the top 10 relevant sentences.

Herein we will discuss our method. Next, we will present the results of comparison of the systems and the performance of the new system on the test set, which was evaluated manually by the organizers. Lastly, we discuss the results and limitations of the system.

II. METHODS

Our system has 3 main components: indexing, retrieval, and ranking.

A. Indexing Component

As the system aims to retrieve evidence sentences for BEL statements, we only store and index *informative sentences* from MEDLINE abstracts. We call a sentence *informative* if it contains at least 2 biomedical entities and a relationship between the entities. To identify these sentences, the system relies on Semantic Medline Database (SemMedDB (10)). SemMedDB is a relational database and stores all informative sentences, from MEDLINE abstracts, which are extracted by a rule-based system, as described by Kilicoglu et al (10). The indexing component retrieves all the sentences from SemMedDB and indexes them in a text search engine, called ElasticSearch. Unlike the other 2 components that run for each query (BEL statement), this component only runs once and prepares the index for the system.

In the previous system (BELTracker), we indexed abstracts from PubMed and full text articles available in PubMed Central. In the current index, we do not have full-text article sentences because SemMedDB does not cover full-text articles.

B. Retrieval Component

For a given BEL statement, the system first retrieves the most relevant sentences from the index, which is the responsibility of the second component, retrieval. The retrieval component identifies all the elements (Table I) in the given BEL and uses external and expert-generated resources to find their synonyms and then generates an appropriate ElasticSearch query. The retrieval component in the system is as the same as BELTracker (9), and the only difference is that we utilize a newer version of the resources.

C. Ranking Component

The retrieval component returns at most 1,000 relevant sentences to the given BEL statement. ElasticSearch retrieves these sentences based on appearance of the BEL's elements somewhere in the sentence, and it does not consider any semantic feature in the retrieval process. It is obvious that simply based on co-occurrence of the elements in a sentence, we can not conclude existence of relation between the elements. The third component of the system, the ranking component, investigates existence of relation between the elements and ranks the sentences based on their relevancy to the BEL statement. In order to rank the evidence sentences, 3 classifiers classify them based on existence of any relation between the elements. Here we describe these 3 classifiers and how the ranking component uses the classification results for ranking the sentences.

1. First Classifier: Entity-Entity Classifier

Instead of using co-occurrence of entities (of the BEL statement) in a sentence as the indicator of the existence of a relation between them, we propose to train a binary classifier, called *Entity-Entity* (EE) *classifier*. Each BEL statement (in the training and test data) has at least 2 entities and EE classifier is calculating the likelihood of relation (regardless of type of relation) between the BEL entities in the retrieved sentences by the retrieval component. The instances of EE classifier are sentences containing at least 2 entities; the positive instances showing relation between the entities and the negative instances otherwise. One of the challenges to train the EE classifier is the training data. The training data provided by the organizers contains only positive instances (6,000 sentences and 11,000 BEL statements extracted from them). In order to generate negative instances, we employ distant supervision technique and SemMedDB. There are sentences in SemMedDB, which contain 2 biomedical entities with *co-exist* relation type. This relation type indicates that the rule-based system was not able to identify any specific type of relation between the entities and they only co-occurred in the sentence. We utilized these sentences as negative instances for the EE classifier. In the training process, the name of entities in the sentences are masked and replaced with a general term such as *Entity*. As features for the EE classifier, unigrams, bigrams, and word embedding of terms between entities are used. Since word embedding contains semantic relationship information, adding word embedding to the feature list allows us to move beyond lexical features. We trained embedding on PubMed abstracts.

2. Second Classifier: Function-Entity Classifier

The second classifier aims to calculate the probability of relation between functions and entities in the retrieved sentences. For example, if the given BEL statement is:

cat(HGNC:XIAP) decreases cat(HGNC:CASP9)

there are 2 function-entity relations: *cat-XIAP* and *cat-CASP9*. This classifier, called *Function-Entity* (FE) *classifier*, examines existence of both pairs in the retrieved sentences and calculates the probability of each relation in the sentences. For each function that appeared more than 100 times in the training data, we build a binary classifier.

In order to train FE classifiers, both positive and negative instances are available in the training data. Unigrams and bigrams of surrounding words of the entity (window 3-5) are utilized as the features for FE classifiers. Beside lexical features, we evaluated using word embedding, dependency-based word embedding (11), and abstract meaning representation (AMR) embedding (12) as other features for FE classifiers. In the following sentence, we illustrate how we generate dependency contexts to train dependency-based word embedding. A similar approach was used to generate AMR

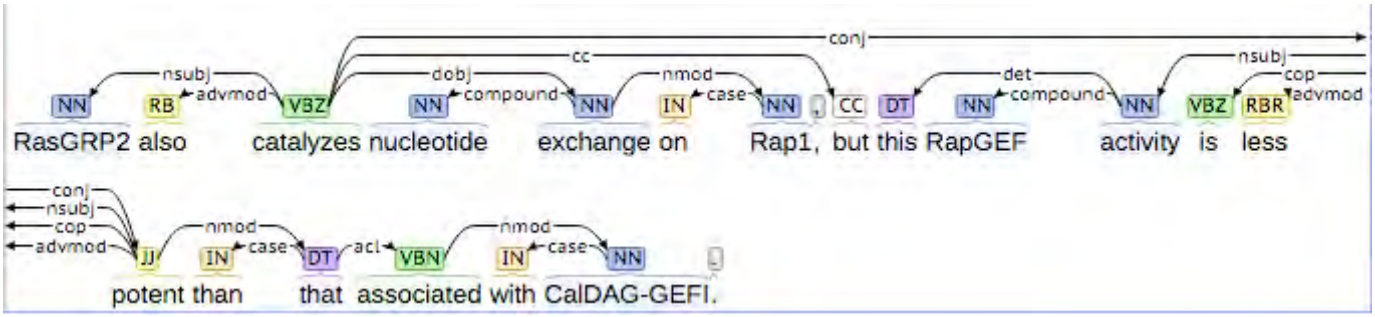


Figure I: Dependency tree for this sentence:
"RasGRP2 also catalyzes nucleotide exchange on Rap1, but this RapGEF activity is less potent than that associated with CalDAG-GEFI" PMID:10918068

embedding from AMR graphs shown in Wang et al (12). Consider this sentence:

"RapGEF activity is less potent than that associated with CalDAG-GEFI."

We generate dependency tree for the sentence(Figure I). Then for each term, we produce dependency contexts for all incoming and outgoing edges. Dependency context for each edge is a combination of *governor of edge*, *direction of edge (inv for incoming edges)*, and *the term on the other side of edge*. Table II contains dependency contexts for first 3 terms in the sentence.

TABLE II. DEPENDENCY CONTEXTS

	Terms		
	RasGRP2	***Also***	***Catalyzes***
Dependency Contexts	1)nsubj_inv_catalyzes	1)advmod_inv_catalyzes	1)nsubj_RasGPR2 2)advmod_also 3)conj_potent 4)cc_but 5)dobj_exchange

3. *Third Classifier: Relation Classifier*

There are 2 types of relationship in the training data: *increase* or *decrease*. We train another binary classifier to categorize the retrieved sentences based on type of relationship. The training data contains both positive and negative instances for this classifier. More details about these classifiers, such as feature list, are explained in our previous work (9).

4. *Final score calculation*

After obtaining probabilities (from the classifiers) for each retrieved sentence, we aggregate the results and calculate a score for each sentence. This score is used for ranking the sentences and selecting top relevant evidence sentences. The score is calculated as follows:

$$Score_{sentence} = W_{EE} * P_{EE} + W_{FE} * P_{FE} + W_{relation} * P_{relation}$$

P represent the probability produced by the classifiers and *W* indicate the weights assigned to each classifier. As we do not have appropriate data to learn each classifier's weight, we assign weights to each classifier based on importance of each element and training data of the classifiers. The weights are: $W_{EE} = 0.4$, $W_{FE} = 0.5$, and $W_{relation} = 0.1$. The FE classifier has the highest weight because the data used to train this classifier has less noise compared to the EE classifier. The relation classifier has a low weight, because we observe that rarely 2 sentences contain all elements of a BEL statement but convey 2 different relationship types. Meaning, if entities and functions of a given BEL statement appear in a sentence, most likely the sentence has the relationship type mentioned in the BEL statement.

III. RESULTS AND DISCUSSION

Using the test set of BioCreative V, we compared mean average precision (MAP) of new and previous systems. Table III shows MAP of the systems and 3 different scenarios (worst, random, best) (scenarios and criteria are described in (9)).

TABLE III. COMPARING MAP OF BOTH SYSTEMS AND 3 SCENARIOS

Criteria	Systems and scenarios				
	Worst	***Random***	***Previous System***	***New System***	***Best***
Full	31.7	46.5	49.0	56.96	74.2
Relaxed	45.9	58.4	62.1	65.05	80.4
Context	55.2	65.7	68.9	73.15	83.5

The results in Table III show the new system obtains higher MAP compared to the previous system; however, there is still a lot of room for improvement (comparing to the best possible MAP for BioCreative V test set). Unfortunately, we are not able to compare precision of the systems, because the evaluation should be done by domain expert, who we do not have access to.

TABLE IV. PRECISION OF EACH RUN

Systems	Criteria	
	Full	***Partial***
Baseline	30.6	45.8
Using the classifiers for ranking	31.6	50.2

For BioCreative VI, the organizers provided 100 BEL statements as a test set and asked the participants to return up to 5 relevant evidence sentences for each BEL statement. We submitted 2 sets of results. In the first run (baseline), we ranked the sentences based on ElasticSearch score (in fact, we did not

engage the ranking component). In the second run, the sentences are ranked using the ranking component. Table IV shows precision of each run for *full* and *partial* criteria. The results show that semantic features can improve the performance of the system. Using semantic features, the system gained 1% and 5% in precision for full and partial criteria, respectively.

There are several limitations in this work. The main limitation is the absence of manually generated training data for the classifiers. The classifiers are training by the dataset which the entities are not annotated manually (keyword search is used to detect and annotate the entities). Using the classifiers is the novelty of this work, but lack of having comprehensive and clean training data forced us to simplify the relations in BEL statements. For example, EE classifiers do not consider relations between more than 2 entities (which is possible in nested BEL statements). As mentioned before, the system is only search sentences in PubMed abstracts, because SemMedDB does not cover full text articles. This leads to another limitation that the system misses evidence sentences which are not in the abstracts.

In the future, we will focus our work on generating cleaner training data for the classifiers and adding sentences from full-text articles available in PubMed Central to the index.

IV. Conclusion

As a participant in BEL track of BioCreative VI, we implemented an information retrieval system to retrieve evidence sentences for BEL statements. Compared to our previous system, BELTracker, we tried to use semantic features such as: word embedding, dependency-word embedding, and AMR embedding in the ranking process. The results showed that these semantic features can improve performance.

References

1. Hoffmann R, Valencia A. Implementing the iHOP concept for navigation of biomedical literature. Bioinformatics. 2005 Jan 1;21(suppl 2):ii252-ii258.
2. Chen H, Sharp BM. Content-rich biological network constructed by mining PubMed abstracts. BMC Bioinformatics. 2004 Oct 8;5:147.
3. Nikitin A, Egorov S, Daraselia N, Mazo I. Pathway studio—the analysis and navigation of molecular networks. Bioinformatics. 2003 Nov 1;19(16):2155–7.
4. Doms A, Schroeder M. GoPubMed: exploring PubMed with the Gene Ontology. Nucleic Acids Res. 2005 Jul 1;33(Web Server issue):W783-786.
5. Ohta T, Matsuzaki T, Okazaki N, Miwa M, Sætre R, Pyysalo S, et al. Medie and Info-pubmed: 2010 update. BMC Bioinformatics. 2010 Oct 6;11(Suppl 5):P7.
6. Ravikumar KE, Wagholikar KB, Li D, Kocher J-P, Liu H. Text mining facilitates database curation - extraction of mutation-disease associations from Bio-medical literature. BMC Bioinformatics. 2015.
7. Fluck J, Madan S, Ellendorff TR, Mevissen T, Clematide S, Lek A van der, et al. Track 4 Overview: Extraction of Causal Network Informationin Biological Expression Language (BEL). In 2015.
8. Fluck J, Madan S, Ansari S, Kodamullil AT, Karki R, Rastegar-Mojarad M, et al. Training and evaluation corpora for the extraction of causal relationships encoded in biological expression language (BEL). Database (Oxford) 2016 Aug 20 [cited 2017 Sep 13];2016.
9. Rastegar-Mojarad M, Komandur Elayavilli R, Liu H. BELTracker: evidence sentence retrieval for BEL statements. Database (Oxford). 2016 May 12.
10. Kilicoglu H, Shin D, Fiszman M, Rosemblat G, Rindflesch TC. SemMedDB: a PubMed-scale repository of biomedical semantic predications. Bioinformatics. 2012 Dec 1;28(23):3158–60.
11. Levy O, Goldberg Y. Dependencybased word embeddings. In: In ACL. 2014.
12. Wang Y, Liu S, Rastegar-Mojarad M, Wang L, Shen F, Liu F, et al. Dependency and AMR Embeddings for Drug-Drug Interaction Extraction from Biomedical Literature. In: Proceedings of the 8th ACM International Conference on Bioinformatics, Computational Biology,and Health Informatics. ACM; 2017. p. 36–43.

A Hierarchical Sequence Labeling System for BioCreative VI BEL task

Jiaxin Liu, Suwen Liu, Yunqi He, Longhua Qian*
NLP Lab, School of Computer Science and Technology, Soochow University, Suzhou, China

***Abstract*—We implemented a hierarchical sequence labeling system for BioCreative VI Track 3 BEL Task 1—extract BEL statements from sentences with or without gold entity annotations. Different from previous systems for BEL task, we mapped sentence-level BEL statements in the BCV 2015 training corpus to the corresponding text segments, thus generating hierarchically tagged training sentences. A hierarchical sequence labeling model was afterwards induced from the training sentences and applied to the test sentences in order to construct the BEL statements. The system achieved an overall F-measure of 22.66% and 10.67% respectively for the test sentences with or without gold entity annotation. The potential of our system is that many advanced machine learning methods can be adopted in the future to further enhance the BEL extraction system.**

Keywords—Biological Expression Language; Dependency Parsing; Alignment Algorithm; Hierarchical Sequence Labeling

I. Introduction

Automatic extraction of biological network information involving proteins, drugs and diseases from biomedical literature is a promising yet demanding task in biomedical text mining. BioCreative VI track 3 task 1 provides a benchmark platform to test various techniques of extracting causal relationships represented in Biological Expression Language (BEL)[1]. Specifically, BEL statements are required to be constructed from sentences in scientific literature.

In the training corpus of BEL track task 1, one BEL statement is annotated corresponding to one sentence or multiple continuous sentences, which means the tags for functions and relationships in a BEL statement cannot be obviously mapped to text segments, therefore training a machine learning model directly from the training sentences and applying it to the test sentences become infeasible. Therefore, previous studies in the BEL task either adopt rule-based methods[2] or apply event extraction/semantic role labeling models induced from other training sets[3] and then transform event/predicate-argument structures to BEL statements. One main drawback of these methods is that the training corpus of BC5 BEL task, which contains roughly 6K informative sentences, is essentially unexplored.

We propose an approach to directly use the BEL training corpus to induce a machine learning model and then apply the model to predicting the test corpus. The main idea is to map a sentence-level BEL statement to the corresponding sentence, i.e., label the text segments with hierarchical tags corresponding to entities, functions and relationships in the BEL statement using an alignment algorithm. After that, sequence labeling models are trained from the tagged sentences and applied to the test sentences in order to reconstruct the BEL statements.

II. System Description And Methods

A. System Framework

Our pipeline system consists of five components: preprocessing, named entity recognition and mapping, parallel corpus construction, training corpus generation and model training/testing. Figure 1 illustrates the framework of the system.

During preprocessing, training sentences are tokenized and BEL statements are normalized. Then, biomedical entities in the training sentences are identified and mapped to the ones in the BEL statements. Next, a parallel corpus is generated between simplified sentences and corresponding BEL statements. A word alignment tool is then applied to the parallel corpus to obtain alignments between words and BEL nodes, from which training examples are generated. Finally, hierarchical sequence labeling models are trained to predict the test sentences and the results are converted to BEL statements.

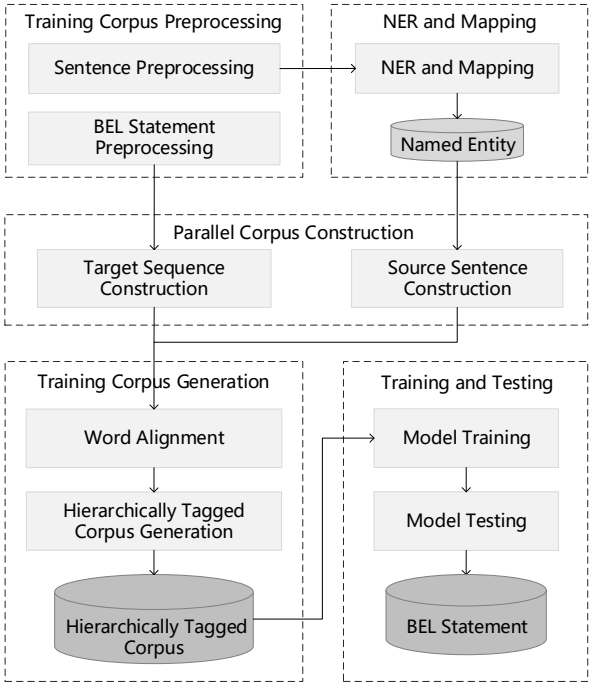


Fig. 1. Framework of the system

* The corresponding author, email: qianlonghua@suda.edu.cn. This research is supported by the National Natural Science Foundation of China [Grant No. 61331011, 61373096 and 61673290].

B. Preprocessing

Preprocessing the training corpus includes two steps. First, we tokenize the sentences using a limited yet effective way. The tokenization here is mainly aimed to facilitate the dictionary-based entity search mentioned in the next subsection. we follow the intuition that a comma followed by a space usually means the end of the sentence, we perform special processing for the hyphens in the sentences. The hyphen in a composite noun will be tokenized if the noun ends with "ed" or "ing", because the past participles and gerunds included in the noun are usually associated with some kind of relationships in scientific literature.

Then, we normalize the BEL statements by resolving the redundancy and inconsistency among them, e.g., there are some cases where two identical statements correspond to the same sentence and other cases where the same entities are involved in two distinct BEL statements. Additionally, in order to facilitate the serialization of the BEL statements, we elevate the hierarchical level of some protein modification functions (including *pmod*, *sub*, and *trunc* etc.) within an entity by reorganizing the entity and the parameters of the function as the child nodes of the function itself. For example, the BEL component "p(HGNC:AKT1, pmod(P, S, 21))" is converted to "pmod(p(HGNC:AKT1), P, S, 21)", thus keeping the functions always above the entities in the BEL statement hierarchy.

C. Named Entity Recognition and Mapping

In the training corpus, entities are given in a BEL statement, but their positions in the sentences are unknown. We adopted three steps in order to maximize the entity recall rate.

- **NER**: first, three NER tools are used to identify biomedical entities, i.e., GNormplus[4] for gene and protein recognition, tmChem[5] for chemical recognition and DNorm[6] for disease recognition. In addition, these tools also link recognized entities to the corresponding entity databases. GNormplus links genes and proteins to Entrez[7], tmChem links chemicals to MESH[8] and CHEBI[9], and DNorm links diseases to MESH and OMIM[10].
- **Mapping**: entity identifiers in the BEL statement, however, are not always the same as the ones recognized by the NER tools, so the second step is to map the latter into the former. Protein ids are consistent across Entrez, HUGO and MGI, so no conversion is needed. Recognized chemical ids are converted to CHEBI ids in terms of their normalized names. Recognized disease ids are discarded if they are linked to OMIM since conversion from OMIM to MESH often leads to loss of information.
- **Dictionary search**: although the three tools achieve the state-of-the-art performance in recognizing biomedical entities, there are still a number of entities in the BEL statement unrecognized, particularly for biological processes. Therefore, we finally performed a dictionary-based entity search for the remaining entities in the BEL statement. The dictionary consists of symbols and synonyms from five entity lists provided by the organizer, i.e., MGI, HUGO, CHEBI, MESHD and GOBP. The matching is based on edit distance and the word sequence with minimal distance to the dictionary items is recognized as the correct one.

For the test corpus in stage 1, the basic steps are similar to the above, and the difference lies in in the step of dictionary search. Here, the search is based on exact matching. We do not use a distance threshold to cut off the partially matched potential entities since the hard-to-set threshold always brings false positives to the system.

D. Parallel Corpus Construction

In order to obtain the alignments between entities, functions and relationships in the BEL statement and the words in the sentence, we recast this problem as the word alignment problem between the source language (text sentence) and the target language (serial representation of the BEL statement). The process includes four stages: BEL tree generation, BEL tree unification, BEL tree serialization, and sentence simplification:

- **BEL tree generation**: in order to serialize BEL statements, they are first converted into tree structure. The aforementioned preprocessing of BEL statements can ensure the success of this conversion. For a BEL statement, the relation is taken as the tree root, and then the relation's left/right arguments are converted in their original order into the children of the tree root. This process can be proceeded in a recursive way until a tree is finally generated. For example, BEL statement "a(CHEBI:castanospermine) decreases complex (p(MGI:Asgr2),p(MGI:Pdia3))" can be converted to BEL tree "(decreases (a CHEBI:castanospermine) (complex (p MGI:Asgr2) (p MGI:Pdia3)))" in the LISP form.
- **BEL tree unification**: one sentence may correspond to multiple BEL trees while one tree may also correspond to multiple sentences. Since BEL statements across multiple sentences are extremely difficult to extract, we focus our attention to statements within one sentence. Multiple trees with coordination or independent relationships are unified by inserting an additional node "or" to produce a single tree in order to align with the sentence. For example, two BEL trees "(decreases (p HGNC:FOXP3) (sec (p HGNC:IL8)))" and "(decreases (p HGNC:FOXP3) (sec (p HGNC:IL6)))" can be unified into "(decreases (p HGNC:FOXP3) (sec (or (p HGNC:IL8) (p HGNC:IL6))))".
- **BEL tree serialization**: with the unified BEL tree at hand, it can be easily transformed into a sequence of nodes via preorder traversal. For example, the above tree "(decreases (a CHEBI:castanospermine) (complex (p MGI:Asgr2) (p MGI:Pdia3)))" is serialized as the node sequence "decreases@2 CHEM1 complex@2 GENE1 GENE2" using the serialization scheme[11], where the sign "@*n*" following function or relation nodes mean those nodes have *n* children. This number is used to reconstruct the tree structure from the node sequence without ambiguity. Here entity names are

replaced with placeholders consisting of entity type name plus the order number of the entity in that type.

- **Sentence simplification**: essentially the BEL statement can be regarded as a kind of semantic representation of sentence. Direct alignment between the whole sentence and the BEL tree may produce many unaligned words, therefore, a dependency-based simplification scheme is adopted to simplify the. Stanford parser[12] is used to parse the sentence into a dependency tree and then the words in the minimal subtree containing all the entities in the BEL statement are rendered as the simplified sentence according to their original order in the whole sentence. For example, the sentence corresponding to the above tree, "Preincubation with a low concentration (15 microg/ml) of the glucosidase inhibitor castanospermine prevented the association of H2a to ERp57 but not to calnexin" (PubMed ID: 14978212) can be simplified to "Preincubation with CHEM1 prevented association of GENE1 to GENE2", which conveys concisely the meaning of the BEL statement. Also, the entities are replaced with placeholders in the same way as for BEL tree serialization.

E. Training Corpus Generation

Generating training corpus from the aforementioned parallel corpus follows two steps: word alignment and hierarchical tag generation:

- **Word alignment**: with the simplified sentence as the source language and the serialized BEL tree node sequence as the target language, their alignment can be readily obtained via GIZA++[13], which is a statistical machine translation toolkit that is used to train IBM Models 1-5 and an HMM word alignment model. The only problem is that in order to ensure that entities in the sentence be aligned to the same entities in the BEL node sequence, many pseudo-parallel sentences like "GENE1 → GENE1" are augmented to the parallel corpus. For example, the alignment result of the above node sequence and the simplified sentence can be represented as "Preincubation/ with/ CHEM1/CHEM1 prevented/decreases@2 association/complex@2 of/ GENE1/GENE1 to/ GENE2/GENE2", where in a aligned word pair the left one comes from the sentence and the right one comes from the node sequence. It occurs that some words in the sentence cannot be aligned to any node in the sequence.

- **Hierarchical tag generation**: based on the alignment result between the nodes in the BEL statement and the words in the sentence, a bottom-up labeling approach is used to annotate the sentence with tags corresponding to BEL nodes layer by layer. The lowest level is for entity and other parameters (such as P, S, or numbers for *pmod*), the immediate upper level (function nodes) is annotated for the text segment spanning between the word aligned to the function node and the words covered by the function node. Finally the top node (the relationship node) is reached and its text span is determined. Take the above sentence as an example, it is annotated as "[[CHEM1]$_{CHEM}$ prevented [association of [GENE1]$_{GENE}$ and [GENE2]$_{GENE}$]$_{complex}$]$_{decreases}$", where a subscript denotes the node type corresponding to the text span enclosed by the pair of brackets.

It should be noted that when dealing with the test sentences, the gold entities are not given in stage 1, and we do not know how many of automatically recognized entities are involved in the potential BEL statements, hence we regard the whole sentences as the test examples. In stage 2, however, the entities along with their positions in the sentences are given in advance, so we can derive the simplified test sentences from the original sentences as the test examples using the same method as for the training examples.

F. Training and Testing

We use the open source CRF package--CRF++[14] to train hierarchical sequence labeling models from the training examples. The first-level sequence labeling model is trained on words and entities. When training the k-th level model, we treat the lower k-1 layers as features. In this recursive way we can finally reach the top-level model. If the maximum model level in the training examples is denoted as $L(4)$, then we need to train 4 models.

In every level of training models, the "BIESO" (begin, in, end, single and out) labeling scheme is used to denote token labels. In traditional sequence labeling-based NER, this scheme usually exhibits best performance. The features used in k-th level CRF model include context words and labels in all the lower k-1 levels with window size 5.

In testing stage, we use the L models trained above to label the test examples in the same order as when we train them. Differently from training, when labeling the k-th layer, the labels automatically recognized in the lower k-1 layers are treated as features.

After labeling all the layers, we convert the labeling results into BEL statements. This process is basically the reverse one of training example generation and can be divided into three steps:

- **BEL tree generation**: convert the hierarchical labeling result of the test sentence to the BEL tree structure.
- **Unified tree splitting**: if there is "or" nodes in the tree, separate the tree into multiple subtrees accordingly.
- **BEL statement generation**: convert every tree into a BEL statement, including normalizing entity type names and moving some protein modification functions (*pmod*, *sub* and *trunc* etc.) inside the entities.

III. Results and Discussion

We participated in both stage 1 and 2 of the BioCreative VI BEL task 1, and three runs were submitted for stage 1, but only one run for stage 2. Table 1 reports the BEL extraction performance with automatically recognized entities (stage 1) and gold entities (stage 2). There is no significant difference between different runs for stage 1, so we only present the best one. The table shows that our system achieves 10.67% and 22.66% of F-measures for stage 1 and stage 2 respectively.

TABLE I. BEL STATEMENT EXTRACTION PERFORMANCE WITH AND WITHOUT GOLD ENTITIES

Level	NER-induced Entities			Gold Entities		
	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)
Term	81.01	**41.97**	**55.29**	**98.83**	**82.95**	**90.20**
Function-Secondary	75.00	4.00	7.59	58.82	13.33	21.74
Function	75.00	3.16	6.06	38.89	7.37	12.39
Relation-Secondary	**84.00**	36.68	51.06	96.61	74.67	84.24
Relation	38.64	14.91	21.52	52.94	35.53	42.52
Statement	22.22	7.02	10.67	32.00	17.54	22.66

Generally, the low performance, particularly low recall rate, is mainly caused by cascaded errors induced during different stages:

- **NER in training**: automatically entity recognition from the training sentences is far from satisfaction, particularly for the biological processes which cannot even be called entities in strict sense. Matching these processes from a BEL statement into its corresponding sentence seems infeasible in some cases.
- **Dependency parsing**: although we trained Stanford parser using GENIA corpus specifically designed for biomedical domain, there is still a lot of errors for long sentences in the scientific literature, particularly for coordination conjunctions and PP attachments.
- **BEL tree unification**: when we unify multiple trees corresponding to a single sentence, we only consider coordination and independence relationships among trees while ignoring other relationships. This will reduce the number of the training examples by ~20%.
- **Word alignment**: while we finally generate 2,900 parallel sentences for word alignment, this corpus size is still insufficient for a better alignment compared with millions of parallel sentences in machine translation.
- **Hierarchical sequence labeling**: it is always the case that lower-level models can achieve better performance than higher models due to fewer training examples for the latter. This leads to the decrease in the overall performance.

Stage 1 results: the significant performance difference between system with and without gold entities is due to the errors induced by the NER module. Particularly, we initially introduced an additional rule-based postprocessing step which is not mentioned in Subsection B of Section II. The rule dictates that if these species mentioned in the abstract containing the test sentence are human-related, then all the proteins are mapped to the HUGO list, otherwise they are mapped to the MGI list. This rule causes many erroneous conversions for this test corpus. We have removed the rule for the present, however, our official results in stage 1 were induced under the rule.

Stage 2 results: given the gold entities, the performance of function extraction is still low, the reason is that it is difficult to align keywords denoting functions to the ones in BEL statements, either because the keywords are discarded during sentence simplification or because there doesn't exist any keywords in the sentence at all. Also, there is a dramatic performance decrease (~40%) from relation-secondary level to relation level, the reason is that some informative words can obviously indicate the type of relationship conveyed in the sentence while determining the scope of the relationship is a relatively challenging task, particularly when the entities are far away from the relation-informative words.

IV. CONCLUSION

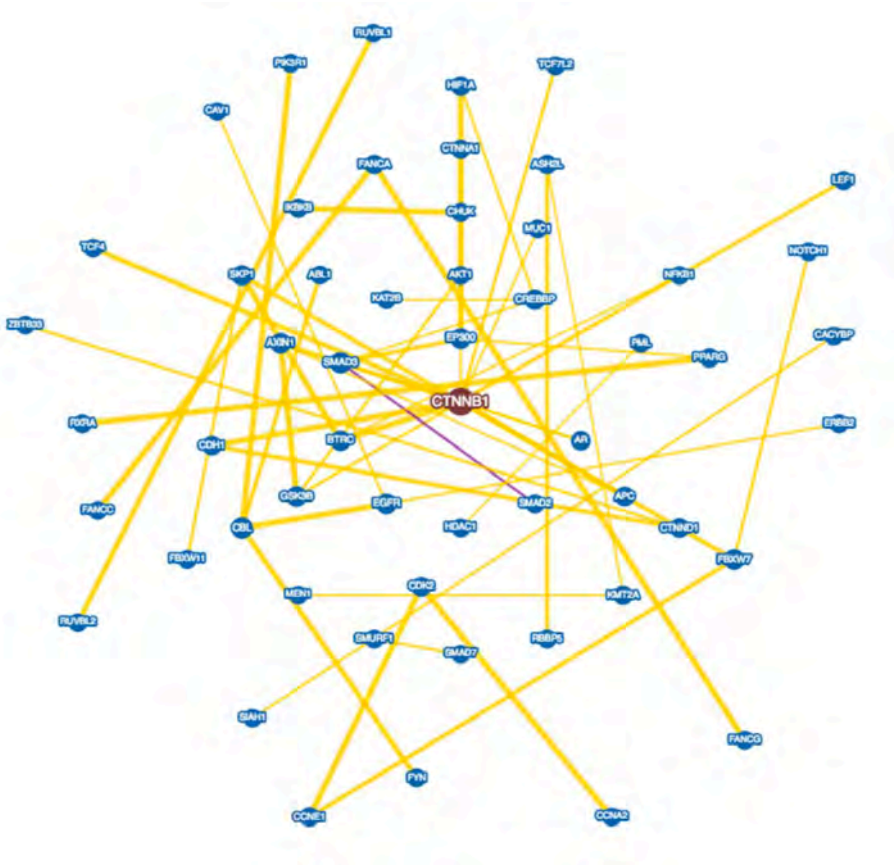
We have implemented a hierarchical sequence labeling system for BEL statement extraction. The main advantage is that we can make use of the training corpus to induce the sequence labeler and then apply it to the test corpus. There are a number of ways to enhance our extraction system in the future, e.g., improve the NER module to recall more entities in the training/test corpus, adjust the BEL tree unification strategy to include more training examples and augment the parallel corpus from other resources etc.

REFERENCES

1. Biological Expression Language（http://openbel.org/）
2. Elayavilli, R. K., Rastegar-Mojarad, M., & Liu, H. (2015). Adapting a rule-based relation extraction system for BioCreative V BEL task. In Proceedings of the fifth BioCreative challenge evaluation workshop. Sevilla, Spain.
3. Choi, M., Liu, H., Baumgartner, W., Zobel, J., & Verspoor, K. (2015). Integrating coreference resolution for BEL statement generation. In Proceedings of the fifth BioCreative challenge evaluatio workshop. Sevilla, Spain.
4. Wei, C. H., Kao, H. Y., & Lu, Z. (2015). GNormPlus: an integrative approach for tagging genes, gene families, and protein domains. BioMed research international, 2015.
5. Leaman, R., Wei, C. H., & Lu, Z. (2015). tmChem: a high performance approach for chemical named entity recognition and normalization. Journal of cheminformatics, 7(1), S3.
6. Leaman, R., Islamaj Doğan, R., & Lu, Z. (2013). DNorm: disease name normalization with pairwise learning to rank. Bioinformatics, 29(22), 2909-2917.
7. Tennant, M. R., & Lyon, J. A. (2007). Entrez Gene: A Gene-Centered "Information Hub". Journal of Electronic Resources in Medical Libraries, 4(3), 53-78.
8. Coletti, M. H., & Bleich, H. L. (2001). Medical subject headings used to search the biomedical literature. Journal of the American Medical Informatics Association, 8(4), 317-323.
9. de Matos, P., Dekker, A., Ennis, M., Hastings, J., Haug, K., Turner, S., & Steinbeck, C. (2010). ChEBI: a chemistry ontology and database. Journal of cheminformatics, 2(S1), P6.
10. Hamosh, A., Scott, A. F., Amberger, J. S., Bocchini, C. A., & McKusick, V. A. (2005). Online Mendelian Inheritance in Man (OMIM), a knowledgebase of human genes and genetic disorders. Nucleic acids research, 33(suppl_1), D514-D517.
11. Li, J., Zhu, M., Lu, W., & Zhou, G. (2015). Improving Semantic Parsing with Enriched Synchronous Context-Free Grammar. In EMNLP (pp. 1455-1465).
12. De Marneffe, M. C., MacCartney, B., & Manning, C. D. (2006, May). Generating typed dependency parses from phrase structure parses. In Proceedings of LREC (Vol. 6, No. 2006, pp. 449-454).
13. Och, F. J., & Ney, H. (2000). Giza++: Training of statistical translation models.
14. Kudo, T. (2005). CRF++: Yet another CRF toolkit. Software available at http://crfpp. sourceforge. net.

Track 4

Mining protein interactions and mutations for precision medicine



Overview of the BioCreative VI Precision Medicine Track

Mining protein interactions and mutations for precision medicine

Rezarta Islamaj Doğan[1], Sun Kim[1], Andrew Chatr-aryamontri[2], Chih-Hsuan Wei[1], Donald C. Comeau[1], and Zhiyong Lu[1]

[1]National Center for Biotechnology Information, National Library of Medicine, National Institutes of Health, Bethesda, USA
[2]Institute for Research in Immunology and Cancer, Université de Montréal, Montréal, Canada

Abstract— **The Precision Medicine Track in BioCreative VI aims to bring together the biomedical text mining community and the biocuration community for a novel challenge composed of two tasks: 1) Triage task focused on identifying scientific articles that describe protein-protein interactions (PPI) being disrupted or significantly affected by the presence of genetic mutations, and 2) Relation extraction task focused on extracting the affected protein pairs. This is a novel challenge for the BioNLP community and, to assist system developers and the assessment of such an endeavor, we created the first large-scale manually annotated corpus of scientific articles that describe such information relevant to precision medicine initiative. The training corpus contained 4,082 articles annotated for triage, of which 598 PubMed articles were annotated for relations. The testing corpus contained 1,464 PubMed articles annotated for both triage and relations. Ten teams worldwide participated in the triage task and sent in results for 22 distinct text mining models. Six teams participated in the relation extraction task and sent in results from 14 different text mining systems. When comparing the text mining system predictions with human annotations, for the triage task, the best F-score was 69.5%, the best precision was 61.6%, the best recall was 97.9% and the best average precision was 72.8%. For the relation extraction task, when we account for similar gene identifiers with HomoloGene database, the best F-score was 37.3%, the best precision was 45.4%, and the best recall was 53.9%. Given the level of participation and team results we find our task to be successful in engaging the text-mining research community, producing a first-of-its-kind, large, manually annotated corpus of scientific articles relevant for precision medicine, and providing the first results of automatically identifying PubMed articles that describe PPI affected by mutations, and extracting the affected relations.**

Keywords—precision medicine, corpus annotation, relation extraction, protein-protein interaction, mutation, information extraction.

I. Introduction

The goal of the BioCreative challenges (1-8) has been to propose tasks that will bring together text mining community and biology researchers in order to foster the development of systems that can help with biologically relevant problems. One such current research area is precision medicine, an emerging approach for disease treatment and prevention that takes into account variability in genes, environment and lifestyle for each person. Because the intricate network of interactions between genes contributes to control cellular homeostasis, differences in interaction stability, although not resulting in any obvious phenotype, can contribute to the development of diseases in specific contexts. Annotating how gene mutations or variations affect the global behavior of the cellular interaction provides additional support to precision medicine efforts.

Such information can be found in the unstructured text within the scientific articles indexed in PubMed (9-14). Specialized curation databases, such as IntAct and BioGRID have been collecting and cataloging knowledge focused on particular areas of biology so that they may enable insights into conserved networks and pathways that are relevant to human health. Expanding their curation efforts into capturing specific sequence-variant-depended molecular interactions may open up new possibilities and enable insights that pertain to precision medicine. To date, no tool is available to facilitate this kind of specific retrieval. The goal of our track is to foster the development of text mining algorithms that specialize in scanning the published biomedical literature and are capable to extract the reported discoveries of protein interactions changing in nature due to the presence of genomic variations or artificial mutations.

II. The Precision Medicine track

The Precision Medicine Track in BioCreative VI is a community challenge that addresses this problem in the form of two tasks:

- Document Triage: Identification of relevant PubMed citations describing mutations affecting protein-protein interactions (PPI).
- Relation Extraction: Extraction of experimentally verified PPI pairs affected by the presence of a genetic mutation.

A. Training and testing datasets

Our first step was the curation of a manually annotated corpus that could be used for the training, tuning and development of text mining algorithms for such a specialized task. Our research on creating and developing our training corpus (15) showed that biomedical literature is ripe with precision medicine relevant information.

TABLE 1 STATISTICS OF THE PRECISION MEDICINE TRACK DATASET

Dataset	Articles	Positive	Negative	Articles with relations	Number of relations
Training	4,082	1,729	2,353	597	752
Testing	1,464	730	734	688	930

We verified this by retrieving articles that were publicly available in expert curated databases, and re-evaluated them for precision medicine purposes. These articles were rich in information of molecular interactions that differ based on the presence of a specific genetic variant, information which could be translated to clinical practice. Moreover, we retrieved articles via state-of-the-art test mining tools that described PPI and contained identifiable sequence variants. Manual curation, again, verified their relevance.

As a result, we released a set 4,082 PubMed abstracts as the Precision Medicine that come from two different sources: curated databases[1] and text mining tool selection. PubMed articles selected from both sources had slightly different, but useful characteristics and as such, novel text mining tools need to use both sources of information for best application in this new domain.

Each article in the precision medicine training dataset was annotated for relevance, and a subset of relevant articles was annotated for relation extraction. Each article annotated for relation extraction contained the relation annotations for the interacting pair of proteins which were affected by mutations identified via their Entrez Gene[2] IDs, and in addition, contained the mention annotations of the interacting genes in the PubMed abstract.

As track participants worked on their text mining models, five BioGRID[3] (16) curators worked on annotating the test dataset. As a result of this effort, 1,464 PubMed articles were annotated by at least two curators for relevance and the interacting genes affected by the presence of a mutation were recorded as interacting pairs in 734 articles. Similarly, to the training set, the relations were described as a pair of Entrez Gene identifiers. Statistics of the dataset are shown in Table 1.

Finally, track participant teams were provided with the raw text of the 1,464 PubMed articles[4] in the test dataset and were asked to return:

- A PubMed article label (relevant/not relevant) for the triage task.
- Pairs of Entrez Gene identifiers, for the relation extraction task.

Each task participant could contribute up to three runs per task and this participation is shown in Table 2.

TABLE 2 PARTICIPATING TEAMS AND THEIR SUBMISSIONS

Team Number	Triage Task	Relation Task
374	3	
375	3	3
379	1	2
391		3
405	1	2
414	3	
418	3	
419	3	
420	1	3
421	3	
433	1	1
Total	**10 teams/22 runs**	**6 teams/14 runs**

B. *Evaluation*

For the final evaluation of the participating runs, text mining predictions were compared to manually annotated data using the standard evaluation procedures: precision, recall, F-score, and average precision. To assist the participants, the organizers set up a group e-mail list where information about the task was posted periodically and several discussions were held.

Organizers also set up a PubTator[5] (17) view, so that track participants could visualize the training data annotations. For the evaluation phase, participant teams were provided with the evaluation scripts to use on their results. The evaluation scripts also served as a self-check to ensure that the data was submitted in the correct format for evaluation. Results, in the forms of system output in BioC (18) format (XML/JSON) and a short paragraph description of the applied method, were submitted via e-mail. For each Results Run, organizers asked participants to submit confidence scores for their predictions, which facilitated the ranking results.

For the Relation extraction task, organizers employed a two-level evaluation:

[1] https://www.ebi.ac.uk/intact/
[2] https://www.ncbi.nlm.nih.gov/gene/
[3] https://thebiogrid.org/

[4] 1,500 PubMed articles were initially released as the test set. However, 36 articles that were difficult to assign labels were later removed, and not used for official evaluation.

[5] https://www.ncbi.nlm.nih.gov/CBBresearch/Lu/Demo/PubTator/

- **Exact Match**: all system predicted relations were checked against the manual annotated ones for correctness. A PPI relation is not defined as directional in this annotation format, so the order of genes is not considered when checking for exact match.
- **HomoloGene [6] Match**: All gene identifiers in the predicted relations and manually annotated data were mapped to common identifiers representing common HomoloGene classes, then all predicted relations were checked for correctness. If the predicted Gene ID and the annotated Gene ID were homologous genes, they were counted as a match.

C. *Benchmarking systems*

For comparison purposes, we developed a baseline method for both Triage and Relation Extraction Tasks. For the Triage Task, we designed a baseline SVM classifier using unigram and bigram features from titles and abstracts of the training corpus (15). For the Relation Extraction Task, we implemented a simple co-occurrence baseline method. The Gene entities in the PubMed articles were automatically recognized using our in-house tools (19-23), and a relation was predicted if two gene entities were found in the same sentence, regardless of whether a sequence variant had been predicted for that article or not.

III. RESULTS

Eleven teams participated in the Precision Medicine Track: ten teams in the document triage task, and six teams in the relation extraction task. Since each team could submit up to 3 runs (i.e. 3 different versions of their tool, or contribute three different methods) for each task, a total of 36 runs were submitted. Participants were from Australia, China, Turkey, Greece, Germany, Portugal and the United States.

For the triage task, we received results of 22 systems (shown in Table 3), 16 of which outperformed our baseline in F-score, 13 on average precision, 2 on precision, and 17 on recall. The best F-score is 69.5%, the best average precision is 72.8%, the best precision is 61.6% and the best recall is 98.0%. The average F-score, average precision, precision and recall were 64.1%, 63.5%, 56.6% and 75.4% respectively.

TABLE 3. TRAGE TASK RESULTS FOR ALL SUBMISSIONS

Team Number	Submission	Avg Prec	Precision	Recall	F1	Data Format
374	Run 1	0.6598	0.5916	0.8315	0.6913	JSON
	Run 2	0.6654	0.5747	0.8699	0.6921	JSON
	Run 3	0.6930	0.6092	0.7836	0.6854	JSON
375	Run 1	0.6808	0.5821	0.7575	0.6583	JSON
	Run 2	0.6688	0.5946	0.6973	0.6419	JSON
	Run 3	0.6750	0.5416	0.8822	0.6712	JSON
379	Run 1	0.4885	0.4622	0.3438	0.3943	XML
405	Run 1	0.5877	0.5478	0.5575	0.5526	JSON
414	Run 1	0.4886	0.4792	0.5849	0.5268	XML
	Run 2	0.5055	0.4957	0.7178	0.5865	XML
	Run 3	0.5098	0.5075	**0.9795**	0.6685	XML
418	Run 1	0.6973	**0.6164**	0.7616	0.6814	XML
	Run 2	0.7083	0.5988	0.8096	0.6884	XML
	Run 3	0.7195	0.6026	0.8205	**0.6949**	XML
419	Run 1	0.5742	0.5718	0.8068	0.6693	XML
	Run 2	0.6010	0.5905	0.5986	0.5946	XML
	Run 3	0.6330	0.5989	0.6096	0.6042	XML
420	Run 1	0.6439	0.5473	0.8712	0.6723	JSON
421	Run 1	0.6687	0.5890	0.8068	0.6809	XML
	Run 2	**0.7284**	0.6112	0.7945	0.6909	XML
	Run 3	0.7103	0.5882	0.8219	0.6857	XML
433	Run 1	0.6617	0.5482	0.8877	0.6778	JSON
BASELINE	-	0.6500	0.6097	0.6356	0.6224	-

[6] https://www.ncbi.nlm.nih.gov/homologene

TABLE 4. RELATIONS TASK HOMOLOGENE RESULTS FOR ALL SUBMISSIONS

System	Submission	Precision	Recall	F1	Data Format
375	Run 1	0.3761	0.3527	0.3640	XML
	Run 2	0.3761	0.3527	0.3640	XML
	Run 3	0.4252	0.3301	0.3717	XML
379	Run 1	0.3057	0.0753	0.1208	XML
	Run 2	0.1135	**0.5387**	0.1875	XML
391	Run 1	0.2360	0.1989	0.2159	XML
	Run 2	0.2343	0.1882	0.2087	XML
	Run 3	0.2378	0.1366	0.1735	XML
405	Run 1	0.0767	0.0247	0.0374	JSON
	Run 2	0.1061	0.0376	0.0556	JSON
420	Run 1	0.4321	0.2978	0.3526	JSON
	Run 2	**0.4544**	0.3161	**0.3729**	JSON
	Run 3	0.4286	0.3097	0.3596	JSON
433	Run 1	0.0803	0.2742	0.1242	JSON
BASELINE	-	0.1460	0.5215	0.2282	

For the relations task, we received results from 14 systems, 8 of which outperformed the baseline based on the F-score, 10 on precision, and 1 on recall, when evaluating on Exact Match.

The HomoloGene evaluation showed a slightly different result: 6 systems outperformed the baseline on F-score, 10 on precision, and only one on recall. The average F-score, precision and recall for the HomoloGene evaluation were 23.6%, 27.7% and 24.5%, respectively. The best F-score, precision and recall were 37.3%, 45.4% and 53.9%, respectively. These results are shown in Table 4.

IV. DISCUSSION AND CONCLUSIONS

Given the level of participation and team results we conclude that the precision medicine track of BioCreative VI was run successfully and is expected to make significant contributions in this novel challenge of mining protein-protein interactions affected by mutations from scientific literature. The training and testing data produced during this effort is novel and substantial in size. Collectively, it consists of 5,546 PubMed articles manually annotated for precision medicine relevance. In addition, the corpus annotations include both text spans and normalized concept identifiers for each of the interacting genes in the mutation-affected PPI relations. We believe that such data will be invaluable in fostering the development of text-mining techniques that increase both precision and recall for such tasks. Another important characteristic is that annotated relations in this corpus are at the abstract level because the majority of such relations are expressed across sentence boundaries.

Participating teams developed systems that specialized in predicting PubMed articles that contain precision-medicine relevant information. Curators at molecular interaction databases will benefit from these text mining systems to select with high accuracy articles relevant for curation. The top achieved recall was 98% and the top achieved precision was 62%. And this is only a first step in this direction. In the future, we plan to build a system that can intelligently merge the results of all individual system submissions with better accuracy.

The relation extraction task on the other hand, showed a somewhat low accuracy. It is to be recognized that this is a very difficult task, as we also showed on the corpus description paper. And, we believe that the accuracy of systems would improve if they were to extract such information from full text. Relation extraction at the abstract level is dependent on accurate entity recognition and correct normalization, as well as the ability to recognize a relation that spans over sentence boundaries, therefore necessitating a system that goes towards abstract-level understanding.

This community effort was designed to foster development of text mining tools that while mining scientific literature could collect information of significant practical value in the clinical practice of precision medicine. The success of the precision medicine endeavor depends on the development of comprehensive knowledge base systems that integrate genomic and sequence variation data, information that lead to tumors and other possible genetic disorders, with clinical response data and outcomes information, as resources for scientists, health care professionals and patients. Leveraging the information already available in scientific literature, and developing automatic text mining methods that facilitate the job of database curators to be able to find and curate such valuable information, is the first step towards this goal.

ACKNOWLEDGEMENTS

This research was supported by the NIH Intramural Research Program National Library of Medicine. Andrew Chatr-aryamontri is supported by National Institutes of Health Office of Research Infrastructure Programs [R01OD010929 and R24OD011194]. We thank the BioGRID database curators: Rose Oughtred, Jennifer Rust, Christie S. Chang, Lorrie Boucher, and Andrew Chatr-aryamontri for annotating the dataset.

REFERENCES

1. Chatr-aryamontri, A., Kerrien, S., Khadake, J., Orchard, S., Ceol, A., Licata, L., Castagnoli, L., Costa, S., Derow, C., Huntley, R., Aranda, B., Leroy, C., Thorneycroft, D., Apweiler, R., Cesareni, G. and Hermjakob, H. (2008). MINT and IntAct contribute to the Second BioCreative challenge: serving the text-mining community with high quality molecular interaction data. *Genome Biol.* **9 Suppl 2**: S5.

2. Hirschman, L., Yeh, A., Blaschke, C. and Valencia, A. (2005). Overview of BioCreAtIvE: critical assessment of information extraction for biology. *BMC bioinformatics.* **6 Suppl 1**: S1.

3. Krallinger, M., Morgan, A., Smith, L., Leitner, F., Tanabe, L., Wilbur, J., Hirschman, L. and Valencia, A. (2008). Evaluation of text-mining systems for biology: overview of the Second BioCreative community challenge. *Genome Biol.* **9 Suppl 2**: S1.

4. Lu, Z. and Wilbur, J. (2010). Overview of BioCreative III Gene Normalization. Proceedings of the BioCreative III workshop, Bethesda, MD.

5. Arighi, C.N., Lu, Z., Krallinger, M., Cohen, K.B., Wilbur, W.J., Valencia, A., Hirschman, L. and Wu, C.H. (2011). Overview of the BioCreative III Workshop. *BMC bioinformatics.* **12 Suppl 8**: S1.

6. Lu, Z. and Hirschman, L. (2012). Biocuration workflows and text mining: overview of the BioCreative 2012 Workshop Track II. *Database : the journal of biological databases and curation.* bas043.

7. Comeau, D.C., Batista-Navarro, R.T., Dai, H.J., Dogan, R.I., Yepes, A.J., Khare, R., Lu, Z., Marques, H., Mattingly, C.J., Neves, M., Peng, Y., Rak, R., Rinaldi, F., Tsai, R.T., Verspoor, K., Wiegers, T.C., Wu, C.H. and Wilbur, W.J. (2014). BioC interoperability track overview. *Database (Oxford).* **2014**.

8. Kim, S., Islamaj Doğan, R., Chatr-aryamontri, A., Chang, C.S., Oughtred, R., Rust, J., Batista-Navarro, R., Carter, J., Ananiadou, S., Matos, S., Santos, A., Campos, D., Oliveira, J.L., Singh, O., Jonnagaddala, J., Dai, H.-J., Su, E.C.-Y., Chang, Y.-C., Su, Y.-C., Chu, C.-H., Chen, C.C., Hsu, W.-L., Peng, Y., Arighi, C., Wu, C.H., Vijay-Shanker, K., Aydın, F., Hüsünbeyi, Z.M., Özgür, A., Shin, S.-Y., Kwon, D., Tyers, M., Dolinski, K., Wilbur, W.J. and Comeau, D.C. (2016). BioCreative V BioC Track Overview: Collaborative Biocurator Assistant Task for BioGRID. *Database.*

9. Singhal, A., Simmons, M. and Lu, Z. (2016). Text Mining Genotype-Phenotype Relationships from Biomedical Literature for Database Curation and Precision Medicine. *PLoS Comput Biol.* **12**(11): e1005017.

10. Caporaso, J.G., Baumgartner, W.A., Jr., Randolph, D.A., Cohen, K.B. and Hunter, L. (2007). MutationFinder: a high-performance system for extracting point mutation mentions from text. *Bioinformatics.* **23**(14): 1862-1865.

11. Cejuela, J.M., Bojchevski, A., Uhlig, C., Bekmukhametov, R., Kumar Karn, S., Mahmuti, S., Baghudana, A., Dubey, A., Satagopam, V.P. and Rost, B. (2017). nala: text mining natural language mutation mentions. *Bioinformatics.* **33**(12): 1852-1858.

12. Horn, F., Lau, A.L. and Cohen, F.E. (2004). Automated extraction of mutation data from the literature: application of MuteXt to G protein-coupled receptors and nuclear hormone receptors. *Bioinformatics.* **20**(4): 557-568.

13. Mahmood, A.S., Wu, T.J., Mazumder, R. and Vijay-Shanker, K. (2016). DiMeX: A Text Mining System for Mutation-Disease Association Extraction. *PLoS One.* **11**(4): e0152725.

14. Saunders, R.E. and Perkins, S.J. (2008). CoagMDB: a database analysis of missense mutations within four conserved domains in five vitamin K-dependent coagulation serine proteases using a text-mining tool. *Hum Mutat.* **29**(3): 333-344.

15. Islamaj Dogan, R., Chatr-aryamontri, A., Kim, S., Wei, C.-H., Peng, Y., Comeau, D.C. and Lu, Z. (2017). BioCreative VI Precision Medicine Track: creating a training corpus for mining protein-protein interactions affected by mutations *Proceedings of the 2017 ACL Workshop on Biomedical Natural Language Processing (BioNLP).*

16. Chatr-Aryamontri, A., Oughtred, R., Boucher, L., Rust, J., Chang, C., Kolas, N.K., O'Donnell, L., Oster, S., Theesfeld, C., Sellam, A., Stark, C., Breitkreutz, B.J., Dolinski, K. and Tyers, M. (2017). The BioGRID interaction database: 2017 update. *Nucleic Acids Res.* **45**(D1): D369-D379.

17. Wei, C.H., Kao, H.Y. and Lu, Z. (2013). PubTator: a web-based text mining tool for assisting biocuration. *Nucleic Acids Res.* **41**(Web Server issue): W518-522.

18. Comeau, D.C., Islamaj Dogan, R., Ciccarese, P., Cohen, K.B., Krallinger, M., Leitner, F., Lu, Z., Peng, Y., Rinaldi, F., Torii, M., Valencia, A., Verspoor, K., Wiegers, T.C., Wu, C.H. and Wilbur, W.J. (2013). BioC: a minimalist approach to interoperability for biomedical text processing. *Database (Oxford).* **2013**: bat064.

19. Wei, C.H., Kao, H.Y. and Lu, Z. (2012). SR4GN: a species recognition software tool for gene normalization. *PLoS One.* **7**(6): e38460.

20. Wei, C.H., Harris, B.R., Kao, H.Y. and Lu, Z. (2013). tmVar: a text mining approach for extracting sequence variants in biomedical literature. *Bioinformatics.* **29**(11): 1433-1439.

21. Wei, C.H., Leaman, R. and Lu, Z. (2016). Beyond accuracy: creating interoperable and scalable text-mining web services. *Bioinformatics.* **32**(12): 1907-1910.

22. Wei, C.H., Kao, H.Y. and Lu, Z. (2015). GNormPlus: An Integrative Approach for Tagging Genes, Gene Families, and Protein Domains. *Biomed Res Int.* **2015**: 918710.

23. Wei, C.H., Phan, L., Feltz, J., Maiti, R., Hefferon, T. and Lu, Z. (2017). tmVar 2.0: Integrating genomic variant information from literature with dbSNP and ClinVar for precision medicine. *Bioinformatics.*

The BioCreative VI Precision Medicine Track corpus

Selection, annotation and curation of protein-protein interactions affected by mutations in scientific literature

Rezarta Islamaj Doğan[1], Andrew Chatr-aryamontri[2], Chih-Hsuan Wei[1], Christie S. Chang[3], Rose Oughtred[3], Jennifer Rust[3], Lorrie Boucher[4], Sun Kim[1], Donald C. Comeau[1], Zhiyong Lu[1], Kara Dolinski[3], and Mike Tyers[2,4]

[1]National Center for Biotechnology Information, National Library of Medicine, National Institutes of Health, Bethesda, MD20894, USA

[2]Institute for Research in Immunology and Cancer, Université de Montréal, Montréal, QC Canada

[3]Lewis-Sigler Institute for Integrative Genomics, Princeton University, Princeton, NJ, USA

[4]The Lunenfeld-Tanenbaum Research Institute, Mount Sinai Hospital, Toronto, ON, Canada

***Abstract*— The Precision Medicine Track in BioCreative VI aims to bring together the biomedical text mining community for a novel challenge: mining the biomedical literature in search of information of value to precision medicine initiatives such as mutations disrupting/affecting protein-protein interactions (PPI). The Precision Medicine track is organized into two tasks: 1) the triage task – focusing on selection of relevant PubMed articles describing PPI affected by mutations, and 2) the relation extraction task – focusing on extracting the interacting gene pairs for the interactions that are affected by the presence of a mutation.**

To support this track with an effective training dataset and limited curator time, the track organizers used a two-staged approach. First, for the creation of the training dataset, the organizers and curators worked on leveraging the information from expertly curated and publicly available PPI databases, augmenting it with a set of articles selected via publicly available state-of-the-art text mining tools. 4,082 PubMed articles were thus carefully reviewed, annotated and released for system development. They contained 1,729 articles labelled positive for curation, out of which, 597 contained 752 curated relations. The second stage pertained to the creation of the testing dataset, which consisted of 1,464 PubMed articles, previously not curated in any of the known PPI databases. These articles were highly likely to describe PPI and sequence variants according to several text mining tests. Each article in the testing dataset was annotated by at least two curators, for relevance relation extraction. Five BioGRID annotators participated and reviewed more than 600 articles each. The testing set contained 730 articles labelled positive for curation, out of which, 688 articles contained 930 curated relations. We detail here the data collection, manual review and annotation process. We give a report on the precision medicine track corpus characteristics. This analysis will provide useful information to developers and researchers for comparing and developing innovative text mining approaches for the BioCreative VI challenge and other Precision Medicine related applications.

Keywords—corpus creation, manual annotation, protein-protein interaction, mutation, relation extraction, information extraction.

I. INTRODUCTION

Biological knowledgebases, such as BioGRID[1], play an increasingly important role in the scientific community due to the curated, summarized and computable knowledge extracted from the literature by expert curators (1, 2). However, their ability to keep up with the growth of biomedical literature is under scrutiny (3). BioCreative (4-13) has traditionally aimed to bridge the gap between the text mining community and biological database curators by fostering development of text mining tools that have practical applications in extracting with high accuracy biological information from unstructured text.

Precision medicine[2] is the emerging approach of disease-treatment that revolves around the idea that a treatment plan is more effective when it takes into account a patient's individual genetic code and the environment they live in. The practice of precision medicine will only be possible with the establishment of databases that integrate the information of genes and mutations with their corresponding biological function. Such knowledgebases will be available for healthcare providers to reference in order to understand the clinical implications of each patient's genetic makeup. The first step towards this goal calls for development of novel text mining tools that can facilitate such an intricate curation processes, increasing accuracy, coverage, and productivity.

To date, there are no available text mining tools that facilitate the specific retrieval of such information which continues to remain buried in the unstructured text within the biomedical literature. The goal of the Precision Medicine Track of

[1]https://thebiogrid.org/

[2] https://syndication.nih.gov/multimedia/pmi/infographics/pmi-infographic.pdf

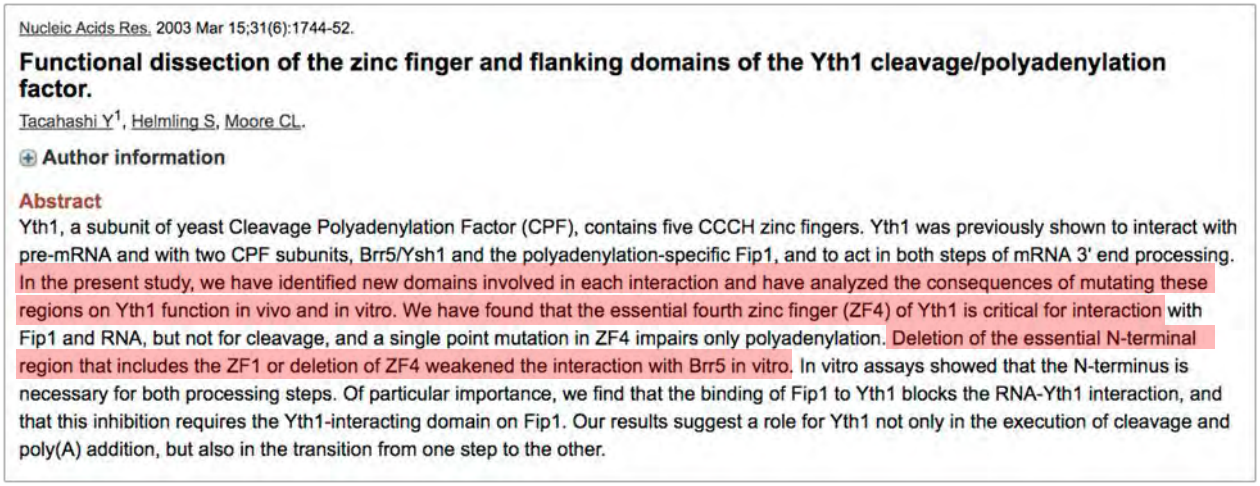
BioCreative VI is to foster the development of text mining algorithms that specialize in scanning the published biomedical literature and to extract the reported discoveries of protein interactions changing in nature due to the presence of genomic variations or artificial mutations. To achieve this goal, we designed the track as a combination of two text mining tasks:

- Document Triage: Identification of relevant PubMed citations describing mutations affecting protein-protein interactions. Figure 1 shows a relevant PubMed article for this purpose, and the highlighted sentences signify to curators that this article describes experimental evidence that the interaction is affected by mutation(s).
- Relation Extraction: Extraction of experimentally verified PPI pairs affected by the presence of a genetic mutation

In order to support this task, we designed and organized an annotation effort that produced a novel corpus. This dataset contains expert human annotations of PPI affected by mutations, as described in the scientific literature. In order to overcome the biggest challenge in building specialized corpora, that of limited reviewer time, we followed several strategies that allowed us to maximize this valuable resource:

First, we built a training dataset consisting of 4,082 PubMed articles, as described in (14). Next, we brought together five BioGRID curators who manually annotated 1,500 PubMed articles for relevance and interacting pairs of proteins that were affected by genetic mutations. Each paper was annotated by at least two curators.

In this manuscript, we describe the process of creating this valuable resource, its manual annotation, annotation guidelines, and inter-annotator agreement. Moreover, we describe how the training and testing datasets complement each other in a rich corpus to test and develop automatic methods for predicting genetically affected protein-protein interactions.


Nucleic Acids Res. 2003 Mar 15;31(6):1744-52.

Functional dissection of the zinc finger and flanking domains of the Yth1 cleavage/polyadenylation factor.

Tacahashi Y[1], Helmling S, Moore CL.

Author information

Abstract

Yth1, a subunit of yeast Cleavage Polyadenylation Factor (CPF), contains five CCCH zinc fingers. Yth1 was previously shown to interact with pre-mRNA and with two CPF subunits, Brr5/Ysh1 and the polyadenylation-specific Fip1, and to act in both steps of mRNA 3' end processing. In the present study, we have identified new domains involved in each interaction and have analyzed the consequences of mutating these regions on Yth1 function in vivo and in vitro. We have found that the essential fourth zinc finger (ZF4) of Yth1 is critical for interaction with Fip1 and RNA, but not for cleavage, and a single point mutation in ZF4 impairs only polyadenylation. Deletion of the essential N-terminal region that includes the ZF1 or deletion of ZF4 weakened the interaction with Brr5 in vitro. In vitro assays showed that the N-terminus is necessary for both processing steps. Of particular importance, we find that the binding of Fip1 to Yth1 blocks the RNA-Yth1 interaction, and that this inhibition requires the Yth1-interacting domain on Fip1. Our results suggest a role for Yth1 not only in the execution of cleavage and poly(A) addition, but also in the transition from one step to the other.

Figure 1 PubMed article relevant for curation. The abstract describes evidence that a protein pair interaction has been affected by a mutation.

II. Corpus Development

The biggest challenge for the organizers of the BioCreative VI Precision Medicine Track was the creation of a high-quality corpus that would serve as a good resource for building automatic algorithms to detect such specialized information.

A possible source for specific PPI information and their related mutations would be the IntAct/Mint database (2), whose curators have had a wide scope when curating protein interactions. Despite the broad coverage and comprehensive curation, such information was not easily retrievable. For this reason, first, our curators selected articles from the IntAct/Mint database that had mutation annotation, and carefully reviewed them and categorized them as relevant/not relevant for the precision medicine track. In addition, we used state-of-the-art text mining methods to select PubMed articles not found in curated databases, that were highly likely to describe protein-protein interactions as well as to contain sequence variations. As a result of this exercise, a set of 4,082 PubMed articles was curated and released as training data to BioCreative VI Precision Medicine Track participants for system development. We described the data repurposing method and text mining triage and manual validation methods that were used to develop this dataset here (14).

The testing dataset, was decided that it would contain previously not annotated articles, and was annotated by five BioGRID curators, with each article being annotated by at least two curators. We describe this process below.

A. *Annotation Guidelines*

The corpus annotation started with a simple exercise for which every PubMed article was categorized based on these questions:

- Does this article describe experimentally verified protein-protein interactions?

- Does this article describe a known disease mutation or a mutational analysis experiment?
- Are the database curated PPI pairs for this article mentioned in the abstract?
- Is the PPI affected by the mutation?

Then, based on the above annotations, articles were carefully categorized as 1) Positives – articles specifically describing PPI influenced by genetic mutations, or 2) Negatives – a category which comprises articles describing both PPIs and genetic variation analysis with no inference of relation between them, articles containing PPI but no mutations, articles containing mutations but no PPI, and articles mentioning neither.

Another important point of consideration was that the information needed to be present in the abstract. Database curators always look for curatable information in the full text. However, the triage process is often conducted on the article's abstract. Thus, for an article to be labelled positive for curation, the title or abstract had to contain a statement of evidence describing in no ambiguous terms that the interaction between a pair of proteins had been affected by the presence of a genetic mutation. The degree of the effect was not annotated.

For the relation extraction task, the interacting proteins needed to be named in the title or abstract, but the name or description of the specific sequence variant was not required. This degree of specificity is unlikely to be found in the abstract, although the information would be present in the full text. Given the condition that the interacting pair needed to be named, it is possible that an article could be labelled positive for the triage task, but not be eligible for the relation extraction task.

Furthermore, protein-protein interactions could be physical interactions, biochemical reactions, self-interactions and/or aggregations. Examples of molecular interactions which were not considered for the relation extraction task are: protein complexes, cell-organelle interactions, and colocalizations. It was also possible that an abstract could describe experimentally verified PPI, as well as include mutations mentions, but the two events were not related. All such articles were labelled negative.

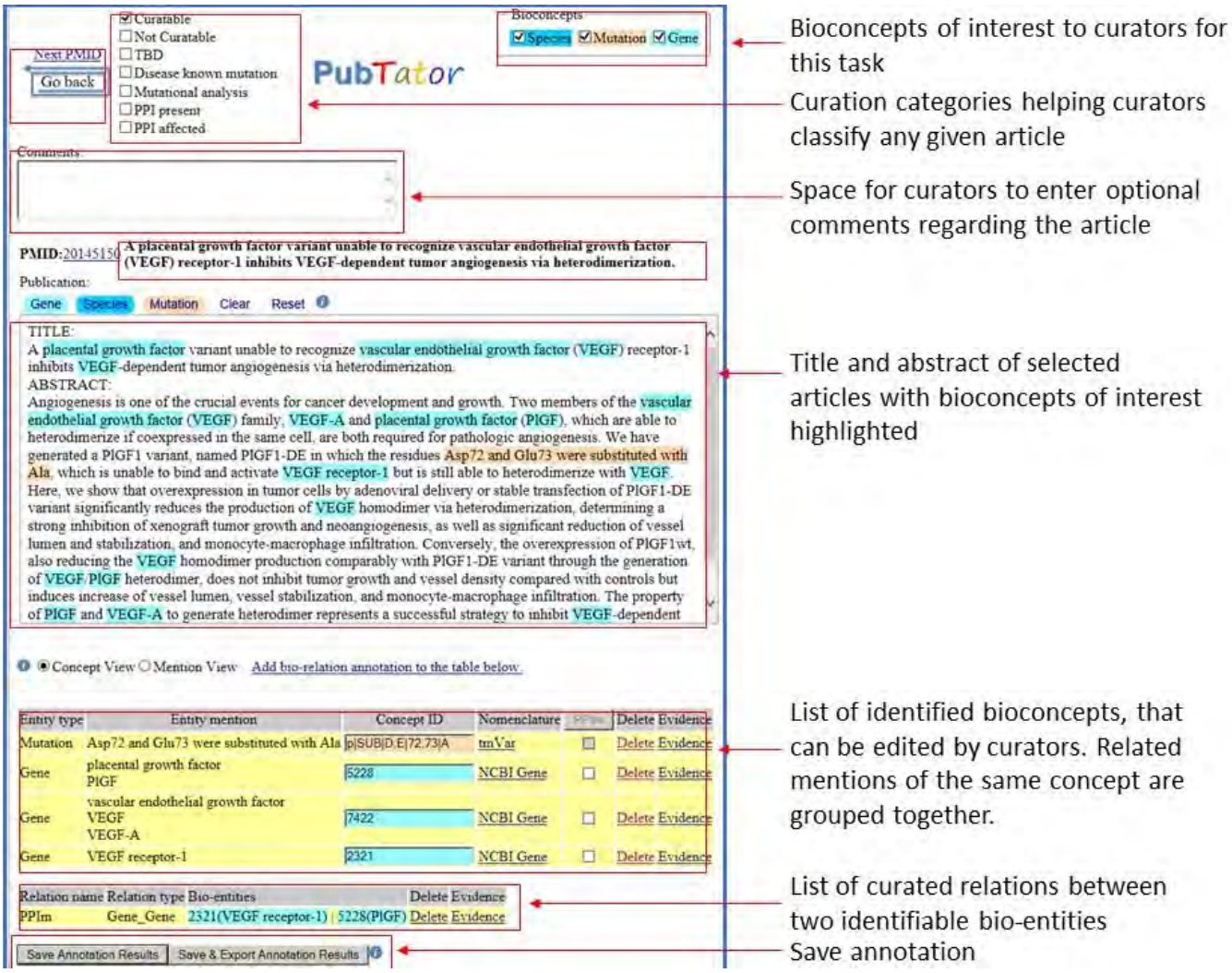


Figure 2 PubTator curation view customized for the BioCreative VI Precision Medicine task. Bioconcepts of interest are: gene names, mutations and species. Automatic detection of these concepts can be turned on and off to help curators.

Regarding mutations, they could be deletions, point mutations and possibly allelic variations. In addition, all mentioned mutations were considered, whether disease mutations or synthetic ones. Often, the mutation was not explicitly mentioned, however if mutational analysis occurred, then the article was curated. Finally, article curation was not limited to any species.

B. Annotation Process

The training data exercise helped collect a set of 4,082 articles which were annotated and distributed for system development. For the testing dataset, we wanted to use PubMed articles that had not been annotated before, and also that could be relevant to curation interests at BioGRID. Starting with a curator-designed comprehensive PubMed query to select articles, which returned 1.4 Million articles, we applied several text mining filters to be able to rank them according to their relevance to the task. Our approach used two well-known publicly available text mining tools: PIE the search (15) and tmVar (16, 17). PIE the search is a web service that ranks PubMed articles based on their probability of describing protein-protein interactions. This algorithm was the winner of BioCreative III ACT competition (5). tmVar is another text mining tool that is used to recognize sequence variants in PubMed literature.

Using these methods, we narrowed down the set of 1.5 Million to 5,000. Given the limited curator time, we randomly selected a set of 1,500 PubMed articles, whose PIE score distribution matched the distribution of scores of the training set, and did two tests of randomly annotating sets of 100 articles to estimate the ratio of relevant to non-relevant articles. After all of these conditions were satisfied, the set of 1,500 PubMed articles was decided upon and the annotation phase of the testing dataset was ready to begin.

All articles in the testing dataset were distributed and randomly assigned to pairs of curators for annotation. The annotation process took place in three phases:

- **Phase 1.** All five curators worked on a set of 20 articles. They spent one week reading and annotating the articles independently, and one week discussing their decisions, for both positively labelled articles and negatively labelled ones. This was the phase where the annotation tool was also adapted to fit curator needs.
- **Phase 2.** Two sets of 100 articles were assigned to three curators at a time. They worked independently for ten days, and then used ten more days to discuss their decisions in groups of three. During this phase, the rules of relation extraction were refined.
- **Phase 3.** The remaining articles, divided into sets of 100, were randomly assigned to pairs of curators. Each pair of curators worked on average for a period of 10 days to curate each set. When both curators were finished with a set, a detailed annotation comparison document was generated, and the curators had independent meetings to review and come to a common agreement. The annotation comparison sheets were used for computing the inter-annotator agreement.

Curators and organizers met weekly to discuss the corpus annotation issues, tool features and report on the progress.

C. Annotation Tool

An annotation portal was built based on PubTator as shown in Figure 2. Testing data was distributed among five curators who accessed the system through private accounts via this system. The system allowed for the organizers to collects multi-annotations for each article and compute annotation comparisons.

When a curator clicks on an article, they view the screen as shown in Figure 2. The tool gives curators the capability to benefit from text mining tools that are specialized in gene/protein, mutation and species identification (mention and normalization). They could easily navigate to the next article, or go to PubMed for more information. They could keep notes on each article. The title and abstract for each article are displayed in one screen, and should the text mining tools be selected, the predicted entities are shown highlighted on the screen. In addition, the predicted list of entities is listed in a table below the abstract. Curators could edit this table to adjust problems. They could annotate from scratch, by highlighting the text mention of interest and selecting the category appearing above the annotation box. Completed annotations could be reviewed, deleted and or edited.

Curators could use the tool to annotate a relation by selecting the entities of interest from the list of bioconcepts in the entity table and clicking on the relation button. The annotated relation then, would get listed in the relations table, shown at the bottom of the screen. Relations could also be edited further as needed. Annotations could be saved, and also exported or downloaded locally.

D. Inter-annotator Agreement

We computed the degree of agreement between pairs of annotators for every set of 100 PubMed articles that was annotated, and then found the average of all sets. For the triage task, on average, our annotators were in agreement for 82% of the articles. The number of articles to be reviewed for classification purposes ranged between 3 and 19 for each set of 100. For each set, on average 2 or 3 articles were difficult to assign a clear label. These were ultimately removed and not used for the official evaluation of the Triage task.

The detailed comparison annotation documents showed that, for each set of 100, on average, 41 articles were marked positive, 42 articles were marked negative and the rest needed to be reviewed to resolve any discrepancies. Of the positive articles, for a typical set of 100 articles, on average, 23 articles needed to be reviewed for the curators to come to a consensus on relation extraction.

TABLE I. STATISTICS OF THE PRECISION MEDICINE TRACK DATASET

Dataset	Articles	Positive	Negative	Articles with relations	Number of relations
Training	4,082	1,729	2,353	597	752
Testing	1,464	730	734	688	930
Total	5,546	2,459	3,087	1,285	1,682

TABLE II. TRIAGE TASK RESULTS OF THE BASELINE SYSTEM

	Avg. Prec.	Precision	Recall	F1
10-fold CV (training data)	0.7225	0.6891	0.6260	0.6561
Testing data	0.6500	0.6097	0.6356	0.6224

TABLE III. RELATION EXTRACTION TASK RESULTS OF THE BASELINE SYSTEM (HOMOLOGENE EVALUATION)

	Precision	Recall	F1
Training data	0.1650	0.4753	0.2449
Testing data	0.1460	0.5215	0.2282

Relations mismatch could be categorized as follows:

- The two curators had picked the same interacting mentions, however, they had normalized them to two different GeneIDs.
- One of the curators had marked additional relations.
- The two curators had marked different interactions, which shared a gene.
- One of the curators, or both, had specifically marked the article for further discussion.

E. *Corpus Characteristics*

As shown in Table 1, the Precision Medicine Track dataset is a large dataset of 5,546 PubMed articles, manually labelled for triage and relations of PPI affected by mutations. In the training dataset, the relations were repurposed from the previous PPI annotations in the IntAct/Mint databases. The testing dataset was richer in relations, since more curator time was devoted to their extraction. As a collection, the Precision Medicine Track dataset contains 1,285 articles annotated for relations with 1,682 total relations.

F. *Benchmark results and corpus use*

A baseline SVM method was designed using unigram and bigram features from titles and abstracts of the training corpus. Results are detailed in Table 2. For the Relation Extraction Task, we implemented a simple co-occurrence baseline method, as shown in Table 3. The Gene entities were automatically recognized using our in-house tools (17-19). The co-occurrence method considered every sentence that contained two gene mentions and predicted a relation between them. Predictions of sequence variants were not considered for this baseline. HomoloGene evaluation, considered whether the curator's annotated gene in the relation and the predicted gene were homologous genes.

This dataset was used for the BioCreative VI Precision Medicine Task. The results of twenty-two systems were submitted for the Triage Task and the results of fourteen systems were submitted for the Relation Extraction Task. This is an indication of the necessity of developing this dataset. We anticipate that more systems will use the released corpus in the future.

III. CONCLUSIONS AND PUBLIC AVAILABILITY

Scientific articles indexed in PubMed contain a vast amount of precision medicine related information, because they often detail experimentally verified protein-protein interactions, which in some cases are affected by differences in sequence variation. Currently, such information can only be extracted by skilled domain expert curators.

The BioCreative VI Precision Medicine Track corpus contains 5,546 PubMed articles and is of high quality. It was curated by five BioGRID curators and each article was annotated by at least two curators, with an inter-annotator agreement of 82%.

By releasing the BioCreative VI Precision Medicine Track corpus, we aim to facilitate the curation of precision-medicine-related information available in published literature. This corpus fosters the development of innovative text mining algorithms that may help database curators in identifying molecular

interactions that differ based on the presence of a specific genetic variant, information which could be translated to clinical practice.

In addition, this dataset may provide important insights on 1) understanding the specific biological information in the unstructured text that may be relevant for precision medicine purposes, and 2) the best practices for designing automatic computational methods that can extract such information.

The BioCreative VI Precision Medicine training corpus is available from the BioCreative website for the scientific community.

ACKNOWLEDGEMENT

This research was supported by the NIH Intramural Research Program National Library of Medicine and by grants from the National Institutes of Health R01OD010929 (to M.T. and K.D.) and 3OT3TR002026-01S1 (sub-award to M.T., Principal Investigator, S. Huang, Institute for Systems Biology, Seattle WA).

REFERENCES

1. Chatr-Aryamontri, A., Oughtred, R., Boucher, L., Rust, J., Chang, C., Kolas, N.K., O'Donnell, L., Oster, S., Theesfeld, C., Sellam, A., Stark, C., Breitkreutz, B.J., Dolinski, K. and Tyers, M. (2017). The BioGRID interaction database: 2017 update. *Nucleic Acids Res.* **45**(D1): D369-D379.

2. Orchard, S., Ammari, M., Aranda, B., Breuza, L., Briganti, L., Broackes-Carter, F., Campbell, N.H., Chavali, G., Chen, C., del-Toro, N., Duesbury, M., Dumousseau, M., Galeota, E., Hinz, U., Iannuccelli, M., Jagannathan, S., Jimenez, R., Khadake, J., Lagreid, A., Licata, L., Lovering, R.C., Meldal, B., Melidoni, A.N., Milagros, M., Peluso, D., Perfetto, L., Porras, P., Raghunath, A., Ricard-Blum, S., Roechert, B., Stutz, A., Tognolli, M., van Roey, K., Cesareni, G. and Hermjakob, H. (2014). The MIntAct project--IntAct as a common curation platform for 11 molecular interaction databases. *Nucleic Acids Res.* **42**(Database issue): D358-363.

3. Hirschman, J., Berardini, T.Z., Drabkin, H.J. and Howe, D. (2010). A MOD(ern) perspective on literature curation. *Mol Genet Genomics.* **283**(5): 415-425.

4. Wu, C.H., Arighi, C.N., Cohen, K.B., Hirschman, L., Krallinger, M., Lu, Z., Mattingly, C., Valencia, A., Wiegers, T.C. and John Wilbur, W. (2012). BioCreative-2012 virtual issue. *Database (Oxford).* **2012**: bas049.

5. Krallinger, M., Vazquez, M., Leitner, F., Salgado, D., Chatr-aryamontri, A., Winter, A., Perfetto, L., Briganti, L., Licata, L., Iannuccelli, M., Castagnoli, L., Cesareni, G., Tyers, M., Schneider, G., Rinaldi, F., Leaman, R., Gonzalez, G., Matos, S., Kim, S., Wilbur, W., Rocha, L., Shatkay, H., Tendulkar, A., Agarwal, S., Liu, F., Wang, X., Rak, R., Noto, K., Elkan, C. and Lu, Z. (2011). The Protein-Protein Interaction tasks of BioCreative III: classification/ranking of articles and linking bio-ontology concepts to full text. *BMC bioinformatics.* **12**(Suppl 8): S3.

6. Lu, Z. and Wilbur, J. (2010). Overview of BioCreative III Gene Normalization. Proceedings of the BioCreative III workshop, Bethesda, MD.

7. Leitner, F., Mardis, S.A., Krallinger, M., Cesareni, G., Hirschman, L.A. and Valencia, A. (2010). An Overview of BioCreative II.5. *IEEE/ACM Trans Comput Biol Bioinform.* **7**(3): 385-399.

8. Krallinger, M., Morgan, A., Smith, L., Leitner, F., Tanabe, L., Wilbur, J., Hirschman, L. and Valencia, A. (2008). Evaluation of text-mining systems for biology: overview of the Second BioCreative community challenge. *Genome Biol.* **9 Suppl 2**: S1.

9. Chatr-aryamontri, A., Kerrien, S., Khadake, J., Orchard, S., Ceol, A., Licata, L., Castagnoli, L., Costa, S., Derow, C., Huntley, R., Aranda, B., Leroy, C., Thorneycroft, D., Apweiler, R., Cesareni, G. and Hermjakob, H. (2008). MINT and IntAct contribute to the Second BioCreative challenge: serving the text-mining community with high quality molecular interaction data. *Genome Biol.* **9 Suppl 2**: S5.

10. Hirschman, L., Yeh, A., Blaschke, C. and Valencia, A. (2005). Overview of BioCreAtIvE: critical assessment of information extraction for biology. *BMC bioinformatics.* **6 Suppl 1**: S1.

11. Kim, S., Islamaj Doğan, R., Chatr-aryamontri, A., Chang, C.S., Oughtred, R., Rust, J., Batista-Navarro, R., Carter, J., Ananiadou, S., Matos, S., Santos, A., Campos, D., Oliveira, J.L., Singh, O., Jonnagaddala, J., Dai, H.-J., Su, E.C.-Y., Chang, Y.-C., Su, Y.-C., Chu, C.-H., Chen, C.C., Hsu, W.-L., Peng, Y., Arighi, C., Wu, C.H., Vijay-Shanker, K., Aydın, F., Hüsünbeyi, Z.M., Özgür, A., Shin, S.-Y., Kwon, D., Tyers, M., Dolinski, K., Wilbur, W.J. and Comeau, D.C. (2016). BioCreative V BioC Track Overview: Collaborative Biocurator Assistant Task for BioGRID. *Database.*

12. Doğan, R.I., Kim, S., Chatr-Aryamontri, A., Comeau, D.C. and Wilbur, W.J. (2015). Identifying genetic interaction evidence passages in biomedical literature. BioCreative V Workshop. Seville, Spain. 36-41.

13. Lu, Z. and Hirschman, L. (2012). Biocuration workflows and text mining: overview of the BioCreative 2012 Workshop Track II. *Database : the journal of biological databases and curation.* bas043.

14. Islamaj Dogan, R., Chatr-aryamontri, A., Kim, S., Wei, C.-H., Peng, Y., Comeau, D.C. and Lu, Z. (2017). BioCreative VI Precision Medicine Track: creating a training corpus for mining protein-protein interactions affected by mutations *Proceedings of the 2017 ACL Workshop on Biomedical Natural Language Processing (BioNLP).*

15. Kim, S., Kwon, D., Shin, S.Y. and Wilbur, W.J. (2012). PIE the search: searching PubMed literature for protein interaction information. *Bioinformatics.* **28**(4): 597-598.

16. Wei, C.H., Phan, L., Feltz, J., Maiti, R., Hefferon, T. and Lu, Z. (2017). tmVar 2.0: Integrating genomic variant information from literature with dbSNP and ClinVar for precision medicine. *Bioinformatics.*

17. Wei, C.H., Harris, B.R., Kao, H.Y. and Lu, Z. (2013). tmVar: a text mining approach for extracting sequence variants in biomedical literature. *Bioinformatics.* **29**(11): 1433-1439.

18. Wei, C.H., Kao, H.Y. and Lu, Z. (2012). SR4GN: a species recognition software tool for gene normalization. *PLoS One.* **7**(6): e38460.

19. Wei, C.H., Leaman, R. and Lu, Z. (2016). Beyond accuracy: creating interoperable and scalable text-mining web services. *Bioinformatics.* **32**(12): 1907-1910.

Hierarchical Bidirectional Attention-Based RNN in BioCreative VI Precision Medicine Track, Document Triage Task

Aris Fergadis[1,3], Christos Baziotis[1,2], Dimitris Pappas[2,3], Haris Papageorgiou[3], Alexandros Potamianos[1,3]
[1]School of ECE, National Technical University of Athens, Zografou 15780, Athens, Greece
[2]Department of Informatics, Athens University of Economics and Business, Patission 10434, Athens, Greece
[3]"Athena" Research and Innovation Center, Maroussi 15125, Athens, Greece

***Abstract*— In this paper, we describe our submission to the "Document Triage Task", of the BioCreative VI Precision Medicine Track, in which we ranked *first* among ten teams. The submitted system is a Hierarchical Bidirectional Attention-Based Recurrent Neural Network (RNN). Our approach utilizes the hierarchical nature of documents, which are composed of sequences of sentences, where sentences are composed of sequences of words. We propose a reusable sequence encoder architecture, which is used as sentence and document encoder. The sequence encoder, is composed of a bidirectional RNN, equipped with an attention mechanism, which identifies and captures the most important elements (words or sentences) in a sequence. Furthermore, we argue that the title of the paper itself, usually contains important information, compared to the other sentences of the abstract. For this reason, we propose a shortcut connection, which integrates the title's vector representation, directly to the final feature representation of the document. We leverage word embeddings, trained on PubMed, for initializing the embedding layer of our network. Moreover, our system does not rely on handcrafted features. Furthermore, we train our system end-to-end using back-propagation, with stochastic gradient descent. We make the source code available to the research community[1].**

***Keywords*— Document Classification; Hierarchical Recurrent Neural Network; GRU; Attention Layer**

I. Introduction

The Document Triage Task is one of the two tasks of the Precision Medicine Track in BioCreative VI. The community challenge that addresses is *"identification of relevant PubMed citations describing mutations affecting protein–protein interactions"* (11).

In this paper we describe our submission to this track, which we approach as a document classification task, where we define the document as the concatenation of the title and abstract of the citation. Usually, document (or text) classification is approached with methods that represent documents with sparse lexical features such as bag-of-words, n-grams, words frequencies (term-frequency and/or inverse-document-frequency – tfidf) and other sophisticated handcrafted features. Linear models or kernel methods, such as variants of Naive Bayes and Support Vector Machines classifiers, are used to train models from those features (20).

Recently, deep neural networks have become popular in NLP tasks, because they can learn underlying features automatically. RNNs have shown great results processing text, especially the variants Long Short-Term Memory (LSTM) (10) and Gated Recurrent Units (GRU) (6) in a variety of tasks (2, 22, 23). Baziotis et al. (2) used a two-layer bidirectional LSTM equipped with an attention mechanism for sentiment analysis in Twitter messages. Yang et al. (22) used a Hierarchical Attention Network with GRU units for document-level sentiment analysis and (23) used a Attention-Based Bidirectional LSTM for relation classification. We employ a hierarchical bidirectional GRU network, equipped with attention layers, which generates dense vector representations for each document and uses those representations as features for classification.

[1]https://github.com/afergadis/BC6PM-HRNN

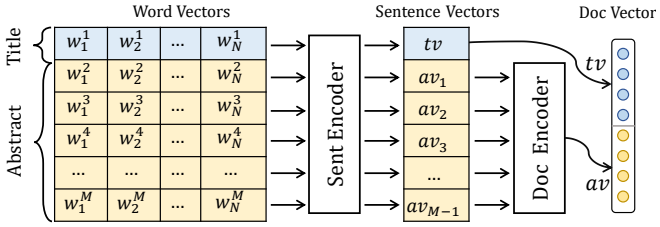


Fig. 1: Overview of our proposed system. *Word Vectors* is an matrix of word embeddings, where M is the maximum number of sentences and N the maximum number of words in a document. tv refers to the *Sent Encoder* representation for the document title and $av_1, av_2, \ldots, av_{M-1}$ to the representations of the abstract's sentences.

II. System Overview

The model we propose is a hierarchical RNN network as shown in Fig 1. We equip the RNN layers with an attention mechanism for identifying the most informative words and sentences in each document. The first level consists of an RNN that operates as a sentence encoder, reading the sequence of words in each sentence and producing a fixed vector representation (sentence vector). Then, a second RNN operates as a document encoder, reading the sequence of sentence vectors and producing the final vector representation for the whole document (title and abstract), which is used as a feature vector for classification. We propose a shortcut connection, which integrates the title's vector representation, directly into the document's vector representation.

A. Text Preprocessing

As a first preprocessing step, we perform sentence segmentation, for splitting the document in it's constituent sentences and tokenization, for splitting the sentences in tokens. We used the Natural Language Toolkit (NLTK) sentence splitter

and word tokenizer to perform these steps (14). We represent each document as a matrix $\mathbf{S} \in \mathbb{R}^{M \times N}$, where M is the maximum number of sentences that a document may have and N is the maximum number of words a sentence may have.

B. Word Embeddings

Word embeddings are dense vector representations of words, capturing their syntactic and semantic information. We leverage the pre-trained word embeddings provided by (18), in order to initialize the weights of the embedding layer of our network. These word embeddings are trained on PubMed articles using word2vec (15) with the skip-gram model and a window size of 5. The dimensionality of the word vectors is 200. Out of vocabulary words, for which we do not have a word embedding, are mapped to a common <UNK> token. We generate the word embedding of <UNK>, by sampling from a uniform distribution, with range $(-0.05, 0.05)$.

C. Recurrent Neural Networks

An RNN processes an input sequentially by performing the same operation $h_t = f_W(x_t, h_{t-1})$ on every element of a sequence, where h_t is the hidden state at time-step t, x_t the input at time-step t, h_{t-1} the hidden state at the previous time-step and W the weights of the network. After reading the whole input sequence, h_t holds a summary of the input, which is used as it's vector representation. Vanilla RNNs suffer for the problem of vanishing gradients (3), which limits their ability to learn long-term dependencies. In order to overcome this limitation, more sophisticated variants of the RNN have been proposed, such as the LSTM (10) or GRU (6), which introduce a gating mechanism in order to ensure proper gradient propagation through the network. In our experiments we used the GRU variant, which has a simpler architecture and achieved better results in our experiments.

D. Attention Mechanism

RNNs have the ability to produce fixed vector representations, for sequences of variable length. The RNN reads each element from the sequence and updates it's hidden state, which by the end, holds a summary of the information contained in the sequence. The output produced at the last time–step, usually a learned non-linear transformation of the hidden state, is used as the vector representation of the whole sequence. However, especially in longer sequences, the RNN does not have the capacity to hold all the relevant information in it's state. This means that the information of certain elements may fade away. In order to amplify the contribution of the important elements in each sequence (words or sentences) we utilize an attention mechanism (6, 8). The attention mechanism assigns a weight to the output from each timestep and the final representation is a weighted combination of all the outputs.

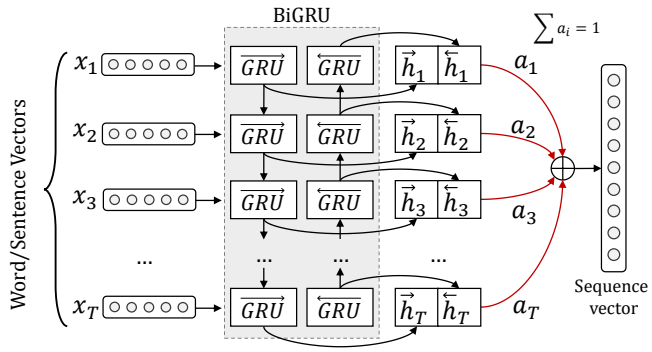


Fig. 2: Architecture of our proposed sequence encoder. The same architecture is used for encoding a sequence of word vectors to a sentence vector (sentence encoder) and a sequence of sentence vectors to a document vector (document encoder).

III. System Architecture

The input to the network is a document, which is composed of a title and an abstract. We perform sentence segmentation and obtain a list of sentences for each document, where the first sentence is the title and the rest of the sentences are the abstract. We feed the list of M sentences to our neural network.

A. Embedding Layer

Given a sentence s^i, which is the i-th sentence of a document, consisting of a sequence of N words $s^i = (w_1^i, w_2^i, \ldots, w_N^i)$, we use an embedding layer to project them to low-dimensional vector space $\mathbb{R}^E$, where E is the size of the embedding layer. We initialize the weights of the embedding layer, with the pre-trained word2vec word vectors (Section II-B).

B. Sentence Encoder

After embedding the words to the low-dimensional semantic space, we use the sequence encoder, in order to obtain a vector representation for each sentence. The sequence encoder consists of a Bidirectional GRU (BiGRU) with an attention layer, which reads the sequence of word vectors of each sentence and produces a sentence vector. The architecture of the sequence encoder, is shown in Fig. 2.

Bidirectional GRU. A GRU takes as input the sequence of word vectors of a sentence and produces a sequence of word annotations (output), $H = (h_1, h_2, ..., h_N)$, where h_i is the hidden state of the GRU at time-step i, summarizing all the information of the sentence up to w_i. We use bidirectional GRU (BiGRU) in order to capture the contextual information of the words from both their left and their right context. A bidirectional GRU consists of a forward GRU $\overrightarrow{f_W}(\cdot)$ that reads the sentence from w_1 to w_N and a backward GRU $\overleftarrow{f_W}(\cdot)$ that reads the sentence from w_N to w_1. We obtain the final annotation for each word w_i, by concatenating the annotations from both directions,

$$h_j^i = \overrightarrow{h_j^i} \parallel \overleftarrow{h_j^i}, \quad j \in [1 \ldots N], \quad h_j^i \in \mathbb{R}^{2S} \tag{1}$$

where $\parallel$ denotes concatenation, $\overrightarrow{h_j^i}$ and $\overleftarrow{h_j^i}$ are the hidden states for the forward and backward GRU respectively at time-step *j* and S the size of the sentence-level GRU layer.

Attention Layer. We use an attention layer in order to identify the most important words in each sentence and enforce their contribution to the final sentence vector. The attention layer, assigns a weight a_j^i to each word annotation h_j^i. The sentence vector sv^i, which is the vector representation of the whole sentence, is computed as the weighted sum of all the word annotations h_j^i.

$$e_j^i = tanh(W_w h_j^i + b_w) \tag{2}$$

$$a_j^i = \frac{exp(e_j^i)}{\sum_{t=1}^{N} exp(e_t^i)}, \quad j \in [1 \dots N] \tag{3}$$

$$sv^i = \sum_{j=1}^{N} a_j^i h_j^i, \quad sv^i \in \mathbb{R}^{2S} \tag{4}$$

where W_w, b_w are the attention layer's weights and bias and sv^i is the vector representation of the *i*-th sentence.

Moreover, we denote the sentence vector of the title as $tv = sv^1$ and the sentence vectors of the abstract as $av_i = sv^i, i \in [2 \dots M]$.

C. Document Encoder

After producing the vector representations for each sentence, we feed them to the document encoder, in order to obtain the final vector representation for the whole document. Notably, we do not feed the vector of the title tv to the sentence encoder, but only the vectors of the abstract av_i. The rationale behind this decision is discussed in Section IV. The remaining sentence vectors, $av_1, av_2, \dots, av_{M-1}$ are feed to the document encoder, in order to get the vector representation of the whole abstract av. The architecture of the document encoder, which is identical to the sentence encoder, is shown in figure 2.

Bidirectional GRU. Similar to the sentence encoder, we use a BiGRU in order to get annotations for each abstract vector av_i, which summarizes the information form the sentences around sentence *i*.

$$h_i = \overrightarrow{h_i} \parallel \overleftarrow{h_i}, \quad i \in [1 \dots M-1], \quad h_i \in \mathbb{R}^{2D} \tag{5}$$

where $\parallel$ denotes concatenation, $\overrightarrow{h_i}$ and $\overleftarrow{h_i}$ are the hidden states for the forward and backward GRU respectively at time-step *i* and D the size of the document-level GRU layer.

Attention Layer. We use an attention layer in order to identify the most informative sentences of the abstract and enforce their contribution to the final vector representation av. The attention layer assigns a weight a_i, to each sentence annotation and we aggregate them by computing the weighted sum of all the sentences annotations.

$$e_i = tanh(W_a h_i + b_a) \tag{6}$$

$$a_i = \frac{exp(e_i)}{\sum_{t=1}^{M-1} exp(e_t)}, \quad i \in [1 \dots M-1] \tag{7}$$

$$av = \sum_{i=1}^{M-1} a_i h_i, \quad av \in \mathbb{R}^{2D} \tag{8}$$

where W_a, b_a are the layer's weights and bias.

D. Output Layer

The final document vector d is computed by concatenating the representations of title and abstract vectors

$$d = tv \parallel av, \quad d \in \mathbb{R}^{2S+2D} \tag{9}$$

The output layer is a fully connected layer with single neuron and a logistic (sigmoid) activation function, which performs the binary classification (logistic regression). It uses the documents vector representation d as feature vector to predict the probability of the two classes.

IV. Hyper-parameters and Training Details

Title Vector Shortcut Connection. Instead of feeding the title vector tv in the sentence encoder with the rest of the sentence vectors (abstract), we create a shortcut connection, by integrating it directly to the final document feature vector d. The reasons for this design decision are twofold. First of all, we argue that the title of a paper contains a lot of important information, which will be diluted if passed in the document encoder with the other sentences, even with the addition of the attention mechanism. By integrating the title vector tv directly into the document feature vector d, we keep the title information intact. Secondly, the sentence encoder is regularized, because (1) it has to produce self-contained representations, which can be used directly as features for classification and (2) it has to produce informative representations as input to the document encoder. This means that the weights of the sentence encoder (sentence BiGRU and Attention layer), receive gradients from both directions: from the output layer and from the document encoder.

Regularization. Neural networks are notoriously prone to over-fitting (13). For this reason we adopt a series of measures, in order to regularize our model.

First, we add Gaussian noise to the input (embedding layer), which limits the amount of information that can be stored in a network (9). This can also be interpreted as a random data augmentation technique, distorting the representation of the words, which means that practically the network never sees the exact same sentence more than once, during training. We add noise by sampling from a zero-mean Gaussian distribution at each batch.

Moreover, we apply dropout to the layers of the network. Dropout randomly disables a certain proportion of the neurons in a layer on each training example (or batch). This means that for each training example, a subpart of the network is trained, which can be thought as a model

ensembling technique. Dropout improves the network's performance because it forces each neuron to learn disentangled features. This way the network learns to recognize the same patterns in multiple ways, which leads to a better model (19). We apply dropout on the embedding layer and on the sentence and document encoders, both on their BiGRU layers and their attention layers.

Many methods have been used to improve stochastic gradient descent such as momentum, annealed learning rates and L2 weight decay. We use Adam (12) optimizer, with the standard deterministic cross-entropy objective function. We add a L2 penalty term (weight decay) to the objective function, in order to discourage large weights and we clip the norm of the gradients at 5 to prevent exploding gradients (16).

As a last step, we perform early-stopping. We stop the training of the network, when the f1-score of the development set stops increasing for a certain number of epochs (17). We decided to monitor f1-score instead of the loss of the development set because its the official evaluation metric used and this way we directly optimize our model for the task.

Hyper-parameter Optimization. The hyper-parameter tuning in neural networks is a very challenging process. In addition to the time consuming training of the neural network, usually we have to tune a lot of hyper-parameters, which are highly correlated (e.g. increasing the number of neurons, changes the optimal dropout rate). As it has been shown in (4), grid search is very inefficient and random search finds consistently better models. However, in our work we adopt the Bayesian optimization approach (5), in order to perform a smart search in the high-dimensional hyper-parameter space. This way, we obtain a set of reasonable hyper-parameters, in a very small number of trials. Table I shows the optimal hyper-parameter values that we obtained.

Experimental Setup. We used Keras (7) to develop our model with Tensorflow (1) as backend. The network was trained on a GTX1070 for approximately 30 minutes.

V. Results

We participated in BioCreative VI Precision Medicine Track, in the "Document Triage Task" as team *418*. According to the official ranking, based on the F1 evaluation metric, our team ranked 1st out of 10 teams, as shown in the Table II. The training dataset, consists of 4082 PubMed citations (abstracts) as shown in (Table III). 1730 are annotated as positive (*relevant*), which means that these abstracts specifically describe protein–protein interaction influenced by genetic mutations. The *test* dataset has a total of 1500 PubMed abstracts.

TABLE I: Hyper-parameter values of our proposed model.

	Layer	Size	Dropout	Noise (σ)
	Embedding	200	0.2	0.2
Sent. Encoder	GRU	150 (x2)	0.3	-
	Attention	1	0.3	-
Doc. Encoder	GRU	150 (x2)	0.3	-
	Attention	1	0.3	-

TABLE II: Official Ranking. Displaying the best F1 score for each team.

Rank	Team #	Avg. Prec.	Precision	Recall	F1
1	418	0,7195	0,6026	0,8205	**0,6949**
2	374	0,6654	0,5747	0,8699	0,6921
3	421	0,7284	0,6112	0,7945	0,6909
4	433	0,6617	0,5482	0,8877	0,6778
5	420	0,6439	0,5473	0,8712	0,6723
6	419	0,5742	0,5718	0,8068	0,6693
7	414	0,5098	0,5075	0,9795	0,6685
8	375	0,6808	0,5821	0,7575	0,6583
9	405	0,5877	0,5478	0,5575	0,5526
10	379	0,4885	0,4622	0,3438	0,3943

TABLE III: Training data set of Document Triage Task.

Positive	Negative	Total
1730 (42%)	2352 (58%)	4082

For our submission we trained our model on 95% of the training corpus and used a held-out 5% as a development set for early-stopping. We adopted this approach in order to ensure that the model used for our submission, did not overfit on the training set. We stopped training when the $F1$ of the development set stopped increasing, optimizing this way our model, directly for the evaluation metric of the task.

VI. Conclusion

We presented a system for the "Document Triage Task" of the BioCreative VI Precision Medicine Track. The key points of the system are: (a) utilizes the hierarchical nature of documents, which are composed of sequences of sentences and sentences are composed of sequences of words, (b) uses a shortcut connection which integrates the document's title directly to the final feature representation of the document, (c) does not utilize any handcrafted feature, and (d) our system is trained end-to-end using back-propagation, with stochastic gradient descent. As future work in this domain, in order to further improve our model, we propose a series of approaches.

Text Preprocessing. Sentence splitting, word tokenization and named entity extraction are crucial steps in text preprocessing. The output of these preprocessing steps has a big impact on the performance of the models. Using a tokenizer tailored for biomedical data, will have the ability to better identify sentence boundaries and difficult tokens such as genes names.

Entity Embeddings. Entities annotations for genes and mutations mentions can be obtained using tools, such as NCBI text-mining web services (21). We plan to utilize these annotations, by representing them as dense vectors which can be appended to the word vectors fed in our model. We hypothesize that this is will improve even further our performance.

Character-level model. An alternative approach would be to design a model, which would operate on the character-level. We argue that such a model, will be able to better model expressions like protein or gene names, reducing the need for careful and laborious text preprocessing.

REFERENCES

1. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I. J., Harp, A., Irving, G., Isard, M., Jia, Y., Józefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D. G., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P. A., Vanhoucke, V., Vasudevan, V., Viégas, F. B., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *Computing Research Repository (CoRR)*, abs/1603.04467.

2. Baziotis, C., Pelekis, N., and Doulkeridis, C. (2017). DataStories at SemEval-2017 Task 4: Deep LSTM with Attention for Message-level and Topic-based Sentiment Analysis. *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, (1):747–754.

3. Bengio, Y., Simard, P., and Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166.

4. Bergstra, J. and Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13:281–305.

5. Bergstra, J., Yamins, D., and Cox, D. (2013). Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In *International Conference on Machine Learning*, pages 115–123.

6. Cho, K., van Merrienboer, B., Gülçehre, c., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *Computing Research Repository (CoRR)*, abs/1406.1078.

7. Chollet, F. and others (2015). Keras.

8. Graves, A. (2013). Generating sequences with recurrent neural networks. *Computing Research Repository (CoRR)*, abs/1308.0850.

9. Hinton, G. E. and van Camp, D. (1993). Keeping the neural networks simple by minimizing the description length of the weights. In *Proceedings of the Sixth Annual Conference on Computational Learning Theory*, COLT '93, pages 5–13. ACM.

10. Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

11. Islamaj, R., Chatr-aryamontri, A., Kim, S., Wei, C.-H., Peng, Y., Comeau, D. C., and Lu, Z. (2017). BioCreative VI Precision Medicine Track : Creating a training corpus for mining protein-protein interactions affected by mutations. *Association for the Computational Linguistics (ACL) Anthology*, pages 171–175.

12. Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *Computing Research Repository (CoRR)*, abs/1412.6980.

13. Lawrence, S., Giles, C., and Tsoi, A. C. (1997). Lessons in neural network training: Overfitting may be harder than expected. *14th National Conference on Artificial Intelligence*, pages 540–545.

14. Loper, E. and Bird, S. (2002). Nltk: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1*, ETMTNLP '02, pages 63–70, Stroudsburg, PA, USA. Association for Computational Linguistics.

15. Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Distributed Representations of Words and Phrases and their Compositionality. *Neural Information Processing Systems (NIPS)*, pages 1–9.

16. Pascanu, R., Mikolov, T., and Bengio, Y. (2013). On the difficulty of training recurrent neural networks. In *International Conference on Machine Learning*, pages 1310–1318.

17. Prechelt, L. (2012). Early stopping—but when? In *Neural Networks: Tricks of the Trade*, pages 53–67. Springer.

18. Pyysalo, S., Ginter, F., Moen, H., Salakoski, T., and Ananiadou, S. (2013). Distributional Semantics Resources for Biomedical Text Processing. In *Proceedings of Languages in Biology and Medicine (LBM), 2013*.

19. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15:1929–1958.

20. Wang, S. and Manning, C. (2012). Baselines and Bigrams: Simple, Good Sentiment and Topic Classification. *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pages 90–94.

21. Wei, C.-H., Leaman, R., and Lu, Z. (2016). Beyond accuracy: creating interoperable and scalable text-mining web services. *Bioinformatics*, 32(12):1907–1910.

22. Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., and Hovy, E. (2016). Hierarchical Attention Networks for Document Classification. *Proceedings of North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2016)*, pages 1480–1489.

23. Zhou, P., Shi, W., Tian, J., Qi, Z., Li, B., Hao, H., and Xu, B. (2016). Attention-Based Bidirectional Long Short-Term Memory Networks for Relation Classification. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 207–212.

Identifying Relevant Literature for Precision Medicine Using Deep Neural Networks

Sérgio Matos and Rui Antunes
Affiliation: DETI/IEETA, University of Aveiro, Aveiro, Portugal

***Abstract*— The aim of precision medicine is to select the best treatments for different patient groups, considering individual variability in genes, environment, and lifestyle. Regarding genetic variability, valuable information about variants and how they affect protein-protein interactions is available in the scientific literature. Extracting and curating this information in an efficient manner requires the application of text mining algorithms. In this paper, we describe deep learning classifiers for selecting, from the literature, documents containing information regarding the impact of genetic mutations on protein-protein interactions. Cross-validation results in the BioCreative VI Precision Medicine task corpus reached a precision of 0,698, recall of 0,735 and f-score of 0,715.**

Keywords—document triage; gene mutations; protein interactions; deep learning; long short term memory networks

I. Introduction

The Precision Medicine (PM) task, part of the BioCreative VI community challenge in biomedical text mining, aimed to evaluate TM approaches and tools for identifying and extracting information regarding the impact of genetic mutations on protein-protein interactions. The challenge consisted of two subtasks, namely document triage and relation extraction. This paper presents our participation in the first subtask, which consisted in identifying PubMed abstracts that mention genetic mutations that affect protein-protein interactions (PPIs).

Document triage is an important first step in assisting literature curation workflows, helping expert curators to focus on papers containing more information relevant to their task. The application of text mining and automatic classification tools for document triage was evaluated in the BioCreative III protein-protein interaction article classification task (PPI-ACT). In this task, the aim was to classify and rank articles relevant for curating protein-protein interactions. The best system was based on a large margin classifier with features derived from gene named entity recognition, MeSH terms and dependency parsing, and reached an area under the Precision/Recall curve (AUC iP/R) of 0,680 and an F-score 0,614.

In recent works, deep learning techniques have been applied succefully in many tasks, most notably in image classification but also when applied to text analysis. Deep learning methods are composed of multiple data transformation layers that apply simple non-linear functions to obtain different levels of representation of the input data (2). Combination of these transformations allow the methods to learn complex classification functions, which gives deep learning its strengths. Another great advantage of such representation learning is that it eliminates the feature engineering effort that is required in more traditional machine learning (2).

For text based tasks, it is necessary to encode the input data in a way that it can be used by the deep network classifier. This is achieved by representing words as embedding vectors of a relatively small dimension, rather than using the large feature space resulting from the traditional one-hot encoding. Word embeddings is a technique that consists in deriving vector representations of words, such that words with similar semantics are represented by vectors that are close to one another in the vector space (3). This way, each document is represented by a sequence of word vectors which are fed directly to the network. Efficient calculation of word embeddings, such as provided by Word2Vec (4), allow inferring word representations from large unannotated corpora. The use of this representation together with deep learning techniques have led to improved results in different NLP tasks, including word sense disambiguation (5), text classification (6), and named entity recognition (7).

II. Methods

We followed a supervised machine learning approach, and evaluated "classical" classifiers against deep learning strategies. In both cases, we used word embeddings to represent the words in the documents.

A. Data

The PM task organizers provided a training set consisting of 4082 citations from PubMed that were manually classified as relevant (1729 documents), that is, containing information regarding the impact of gene mutations on protein-protein interactions, or not relevant (2353 documents).

Apart from this annotated dataset, we exploited the use of the BioCreative III PPI-ACT corpus as additional data. This corpus consists of 12280 Medline abstracts, 2732 of which were annotated as containing PPI information. Although this annotation does not consider the impact of genetic mutations, as is the case of the task considered here, we tried to incorporate this data in a self-learning approach.

B. Word Embeddings

We used the word2vec implementation in the Gensim framework (8) generated word embeddings from the complete MEDLINE database, corresponding to 15 million abstracts in English language. We created six models, with vector sizes of 100 and 300 features and windows of 5, 20 and 50. The models

This work was supported by FCT - Foundation for Science and Technology, in the context of the project IF/01694/2013.

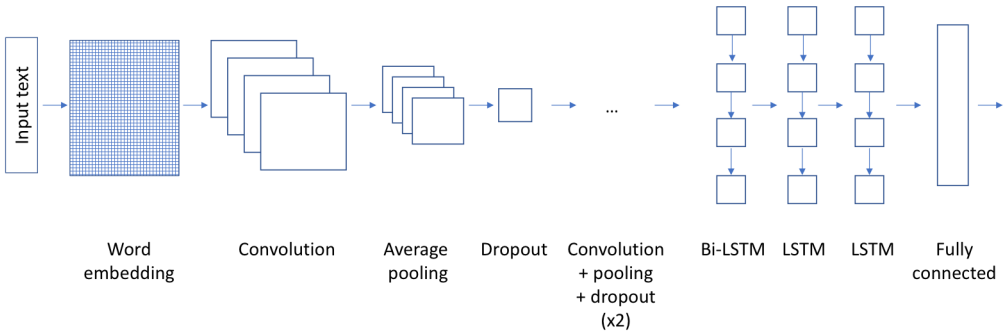


Fig. 1. Deep neural network used in run 3.

contain around 775 thousand distinct words. Based on previous results both in document triage and biomedical word sense disambiguation (9), we used the model with 300 features and window size of 50 in all classifiers tested.

C. Classical Classifiers

We compared three classifiers, namely k-Nearest Neighbors (kNN), Logistic Regression (LR), and Multilayer Perceptron (MLP). These classifiers were selected because they provided good results in previous document classification works. The Scikit-learn library (10) was used for this task.

To obtain the document representation for the classifier, we tokenize the document and obtain the sequence of word vectors by simple look-up in the pre-calculated word2vec model. However, these classifiers are not directly applicable to sequences of distinct length and some form of aggregating these sequences is required. This is commonly addressed by summing or averaging the word vectors, resulting in a single vector representation of the document. We follow a similar approach, but use a weighted average of the word vectors, using as weights the coefficients of a linear regression between the *tf-idf* term-document matrix and the label (0 or 1) of each document in the training set.

D. Deep Learning

We applied different deep learning models based on convolutional and long short term memory (LSTM) layers. Convolutional neural networks (CNN) have been extensively applied in image recognition and classification problems, with very good performance. Various works also demonstrate their application in text classification tasks. Nonetheless, the sequential nature of natural texts can be better modelled by recurrent networks, which contain a feedback loop that allows the network to use information regarding the previous state. LSTM are a special type of recurrent neural networks (RNN) in which a set of information gates is introduced that allow these networks to learn long-term dependencies while avoiding the vanishing gradient problem.

An important consideration when defining and training deep network models is related to overfitting, which means that the network learns the "best" data representation but is not able to generalize to unseen data. Various strategies have been proposed and are commonly employed to address this problem. In our experiments, we applied several strategies to avoid overfitting, namely early stopping, dropout, and regularization. Early stopping looks at the value of the loss in a validation subset and stops the training process when this value stops decreasing. We used 10% of the training data, selected randomly at each run, as validation set. We applied dropout to the output of the embedding and hidden layers so that a random selection of the output tensors is not used for updating the model weights, with the aim of forcing the model to learn a less biased representation of the data. Finally, L2 regularization is applied to the final layer to penalize large weights that could otherwise be assigned to biased input dimensions.

E. Submitted Runs

We empirically tested various network architectures that resulted in our final submitted runs for the task, as described below. All models were trained using the binary cross-entropy loss function and the rmsprop algorithm as optimizer. Models were implemented with the Keras framework (11) with the TensorFlow backend and executed on a machine with 12 CPU cores and 192 GB of memory.

- Run 1

The network architecture for this run consists of an embedding layer, that represents each word in a document by its respective word vector, followed by three convolutional layers with average pooling. Each convolutional layer uses 128 filters with Rectified Linear Unit (ReLU) activation and a kernel size of 3. The output is then connected to a bidirectional LSTM layer with 128 units, and to a final densely connected layer with a sigmoid activation function. L2 regularization with parameter 0.01 was applied in this final layer. A dropout of 0.1 was included after the embedding layer and of 0.2 within the LSTM units, to avoid overfitting as discussed above.

- Run 2

In this run we followed a self-learning approach to incorporate the BioCreative III PPI corpus as additional data. Since this corpus is annotated differently, we first applied the trained model from run 1 to these documents and selected the

ones that were classified with a confidence value higher than 0.90. This equated to adding 9673 documents to the training data. The same network was then re-trained with the combined dataset.

- Run 3

In this run, we applied a deeper network composed of three convolutional layers and three LSTM layers, as illustrated in Fig. 1.

The first layer of the network is the embedding layer, as in the previous runs. A dropout of 0.2 is applied to the embedding vectors, to minimize overfitting. This layer is followed by three convolutional layers with average pooling, with 64 units with ReLU activation and kernel size set to 5. A dropout of 0.4 is applied after each pooling stage. This is then followed by a bidirectional LSTM layer and two unidirectional LSTMs. All three LSTMs are composed of 128 units and use a dropout of 0.2. Finally, a dense layer is applied, with sigmoid activation and L2 regularization with parameter set to 0.01.

III. Results

Table I shows the cross-validation results obtained on the training set. In the case of the classical classifiers, we also show the best f-score obtained when the classification threshold is selected non-blindly.

The best cross-validation results were obtained by the deeper network. This classifier shows an increase of 4 f-score points over the LR classifier operating with an optimized threshold, and of 11 points over the LR and MLP classifiers operating at the default classification threshold.

Comparing to the classical classifiers tested, the deep learning architectures implemented obtained better results in terms of recall and f-score, when the classification threshold was not tuned. When this threshold was 'optimized', the classical classifiers provided a higher recall and, in the case of logistic regression, a higher f-score than the two first architectures. However, since this threshold was selected in a non-blind way, the results may overestimate what would be achieved in a blind dataset.

The use of additional data helped improve the results of the first deep network, although only by a small margin. Nevertheless, this result indicates that careful inclusion of related datasets, when available, can lead to better classification performance.

TABLE I. Cross-validation results on training set

Classifier	*threshold*	*Precision*	*Recall*	*F-score*
kNN (k=99)	0,50	0,618	0,553	0,582
	0,40	0,522	0,845	0,645
LR	0,50	0,674	0,546	0,603
	0,35	0,578	0,798	0,670
MLP	0,50	0,609	0,598	0,603
	0,25	0,554	0,781	0,648
Submitted run 1	-	0,637	0,681	0,651
Submitted run 2	-	0,640	0,692	0,664
Submitted run 3	-	**0,698**	**0,735**	**0,715**

[a]Five-fold cross-validation results on the training set. Different classification thresholds were tested for each classifier. The best results for the MLP classifier were obtained with the word embeddings model of size 300 and window of 5.

References

1. Krallinger,M., Vazquez,M., Leitner,F., et al. (2011) The Protein-Protein Interaction tasks of BioCreative III: classification/ranking of articles and linking bio-ontology concepts to full text. BMC bioinformatics, 12(8), S3.

2. LeCun,Y., Bengio,Y. and Hinton,G. (2015) Deep learning. Nature, 521(7553), 436-444.

3. Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. (2003). A neural probabilistic language model. Journal of machine learning research, 3(Feb), 1137-1155.

4. Mikolov,T., Chen,K., Corrado,G. and Dean,J. (2013) Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

5. Yepes, A.J. (2017) Word embeddings and recurrent neural networks based on Long-Short Term Memory nodes in supervised biomedical word sense disambiguation. Journal of Biomedical Informatics, 73, 137-147.

6. Kim,Y. (2014) Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882.

7. Habibi,M., Weber,L., Neves,M., Wiegandt, D.L. and Leser,U. (2017) Deep learning with word embeddings improves biomedical named entity recognition. Bioinformatics, 33(14), i37-i48.

8. Rehurek,R. and Sojka,P. (2010). Software framework for topic modelling with large corpora. In In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks.

9. Antunes,R. and Matos,S. (accepted). Supervised learning and knowledge-based approaches applied to biomedical word sense disambiguation. Journal of Integrative Bioinformatics.

10. Pedregosa,F., Varoquaux,G., Gramfort,A., et al. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12(Oct), 2825-2830.

11. Chollet,F. et al. (2015) Keras. https://github.com/fchollet/keras

Document Triage and Relation Extraction for Protein-Protein Interactions affected by Mutations

Qingyu Chen[1], Nagesh C. Panyam[1], Aparna Elangovan[1], Melissa Davis[2], Karin Verspoor[1]

[1]School of Computing and Information Systems, The University of Melbourne, Melbourne VIC Australia
[2]Walter and Eliza Hall Institute, Melbourne VIC Australia

***Abstract*—We describe the University of Melbourne READ-Biomed team's participation in the Document Triage and Relation Extraction tasks of the Precision Medicine track of BioCreative VI. For the Document Triage task, we create term lists consisting of terms that are used to describe interactions, mutations, and expected effects on interactions mutations may have. We apply them along with a range of standard bag-of-word features to capture close to 30 features for building classification models. The best model provides a roughly 10% (absolute) increase in F1-score as compared to baseline results, based on 10-fold cross-validation in the training data. For the relation extraction task we use GNormPlus to recognize and normalize gene names. We use two methods, a co-occurrence based method and Support vector machine (SVM) for relation extraction. They achieve 27.4% and 27.2% F1 scores respectively, based on 5-fold cross-validation over the training data.**

Availability—The codes are available from https://biodbqual@bitbucket.org/readbiomed/biocreative-vi.git

Keywords—Document triage; Relation extraction;

I. INTRODUCTION

Text mining can play an important role in precision medicine by supporting automatic extraction of knowledge from ever increasing volumes of scientific literature to facilitate clinical analysis [1]. It has been widely applied to recognize mutations [2] and to extract protein-protein interactions [3]. Nevertheless, little work has addressed the integration of these tasks to specifically identify and extract protein-protein interactions affected by mutations, which is critical in the precision medicine context [4]. The BioCreative VI Track 4 aims to bring community effort to tackle this challenge. The track contains two tasks: the *Document Triage* task, requiring identification of relevant documents describing protein-protein interactions affected by mutations and the *Relation Extraction* task, extracting these specific protein-protein interactions.

We summarize our participation in both tasks. For the Document Triage task, we develop machine learning models leveraging both text mining tools and manually created term lists, achieving a ~10% improvement on F1-score compared to the baseline results. For the relation extraction task, we used a simple rule-based co-occurrence strategy and also attempted SVM with ASM kernel approach, achieving the first rank in the test collection.

The project receives funding from the Australian Research Council through a Discovery Project grant, DP150101550.

II. DOCUMENT TRIAGE TASK

Document triage is the process of identifying documents that are relevant to a user's information needs from a large collection of documents in an efficient manner, which avoids tedious manual document examination [5]. The *Document Triage* task, in the context of the Precision Medicine track of BioCreative VI aims to identify relevant documents describing protein-protein interactions impacted by mutations. We propose a machine learning approach leveraging both standard text mining tools and manually created term lists.

A. Capturing mutations and interactions via BioNLP tools

The notion of relevant documents in this task must satisfy three conditions: a relevant document must (1) mention at least one mutation, (2) describe at least one protein interaction, and (3) indicate that there is some change in the interaction that can be considered to be caused by the mutation. Text mining tools are often used to identify mutations (such as *tmVar* [2] and *EMU* [6]) and interactions (such as *PIE the search* [7]). Thus, as a preliminary step, we apply a number of tools to identify mutations and interactions in documents in the training dataset. The training dataset has 4082 documents; 1729 of them are relevant whereas the rest are not. The process for determining whether those documents are relevant is detailed in the dataset description [4].

TABLE I. DOCUMENTS BEING IDENTIFIED HAVING MUTATIONS

Tool	***Relevant docs*** 1729 total	***Irrelevant docs*** 2353 total
tmVar [2]	965 (55.8%)[a]	1672
EMU [6]	900 (52.1%)	1518
SETH [8]	907 (52.5%)	1541
MutationFinder [9]	881 (51.0%)	1489

a. The figures show that tmVar found 965 relevant documents having mutations, accounting for 55.8% of total relevant documents. Similar results are seen for other tools.

The results show that text mining tools cannot fully capture mutations or interactions as relevant for this task. For example, TABLE I shows the number of documents in which mutations were identified by mutation detection tools. For a document to be labelled as relevant requires that the document includes a reference to a mutation, yet the best text mining tool can only find a specific mutation mention in 56% of these documents, which is consistent with the findings in related studies [10]. Similar results are found for *PIE the search* on

the task of identifying interactions. The main challenge is that many mutations or interactions are mentioned through general references rather than precise descriptions: as an example, "mutagenesis" is the only term describing mutations in the paper PMID:20485264; this general mention is not detected by the tools. This preliminary analysis demonstrates the necessity to leverage text mining tools and term lists for this task.

B. Building term lists for mutations, interactions and impacts

We developed term lists for words or phrases describing mutations, interactions and the impacts on interactions due to mutations. General mutation terms were derived from the Variome corpus [10] and impact terms are selected from Molecular interactions ontologies.[1] The mutation term list contains 22 terms. 13 of these are "strong" terms: *mutation*, *delete*, *mutant*, *insertion*, *substitution*, *-/-*, *+/-*, *polymorphism*, *SNP*, *lesion*, *mutagenesis*, *deleterious* and *variant*. These terms are "strong" in that they are mostly unambiguously used to describe mutations in the literature. In contrast, the remaining 9 terms are "weak": *change*, *exchange*, *damage*, *remove*, *replace*, *disorder*, *deficiency*, *virulence* and *truncation*. While they are used to describe mutations, they may also be used in other contexts; for instance, the term change can be used in "DNA base change" but can also be used in "changes in the distribution". We process strong and weak terms separately when developing features.

The interaction term list contains 30 terms describing interactions frequently in literature, such as *interact*, *complex*, *bound*, *bind*, *regulate*, *kinase*, *acetylation*, *phosphorylation*, and many others. We further create a list of 23 terms that are indicative of the impact a mutation may have on a protein interaction. For example, *increase* is a term often used to describe that the presence of a mutation increases the level of interaction. Other terms include *degrade*, *decrease*, *strengthen*, *enhance*, *reduce*, and *impair*.

C. Leveraging text mining tools and term lists

We use 29 features to build classification models using machine learning. The features can be categorized into document-level mentions, and sentence-level mentions, and are calculated for both raw frequencies and probabilities of different kinds. Representative features are described below.

Document-level features capture mutations and interactions mentioned per document, either by using tools or term lists. The features look for total mentions (total number of mutations identified by text mining tools), unique mentions (total number of unique genes identified by named entity recognition tools) and co-occurrences of mentions (total number of mutations and interaction mentions if both exist).

Sentence-level features are similar to document-level features but focus on individual sentences, e.g., the number of sentences containing both mutation and interactions, identified by either tools or term corpus. We further develop features that represent mutation-impact-interaction triples for sentence bigrams, that is, co-occurrences of mutations and interactions (identified by either tools or terms) exist and the impact terms also exist in every two sentences.

Frequency-based features and probability-based features are calculated at both document and sentence levels. Frequency-based features include the number of sentences containing both interactions and mutations. Probability-based features include the average probability of sentences containing both interactions and mutations, calculated using normalised functions on interaction and mutation frequencies.

TABLE II. REPRESENTATIVE FEATURE ENGINEERING RESULTS

Features	Mean F1 score	Standard deviation
Baseline	0.6405	0.0337
Document-level	0.6850	0.0315
Sentence-level 1-gram	0.7104	0.0333
Sentence-level 2-gram	0.7143	0.0385

Baseline: TF-IDF & Logistic Regression model, Document-level: Baseline & selected document-level features, Sentence-level 1-gram: Document-level & selected sentence-level features, Sentence-level 2-gram: Sentence-level 1-gram & features capturing mutation-impact-interaction triplet, F1 score and standard deviation are based on 10-fold cross-validation.

TABLE III. REPRESENTATIVE MODEL RESULTS

Metrics	Boosting Logistic Regression	SVM
F1 measure	0.7355	0.7097
Average Precision (ranked)	0.7580	0.7479
AUC	0.8306	0.8212

The results are based on 10-fold cross-validation.

D. Experimental results and discussion

We perform feature engineering, train multiple classification models and measure the performance via a range of metrics. The representative results are demonstrated in TABLE II and TABLE III. To process the texts, documents are tokenized and stemmed with stop words removed using NLTK. A baseline model is built using TF-IDF weighted token features and logistic regression. The evaluation using 10-fold cross-validation on the baseline model gives ~64% F1, which is close to the results reported by task organizers [4]. The model is further enhanced by incorporating the features developed above. Document-level and sentence-level features can each increase F1 score by ~4%; the mutation related features and co-occurrence of mutation and interaction related features contribute the most to performance.

We build three models: Logistic Regression with boosting (the results are submitted as Run 1), SVM (Run 2) and Random Forest (Run 3); produce a ranked list, where documents are ranked based on confidence scores; quantify the performance using three metrics: F1-score (summarising balanced precision and recall), Average Precision (measuring the effectiveness of the retrieved ranked list) and AUC (measuring the effectiveness on classifying relevant documents). The results illustrate that Logistic Regression

[1] From www.ebi.ac.uk/ols/ontologies/mi

with boosting gives the best performance: almost 10% F1 improvement compared to the baseline score. Its AP score is over 75% and AUC is over 83%, showing that it has potential to effectively classify relevant documents. SVM results in similar AP and AUC scores, but its F1 score is lower.

III. RELATION EXTRACTION TASK

The goal of the Relation Extraction subtask is to extract protein pairs that interact, where that interaction is impacted by a mutation (a PPIm relation). For instance, the following sentence "*Furthermore, mutagenesis studies indicated that the alpha-helical regions of KIS participate in beta-tubulin binding*", indicates that the interaction between the proteins "*KIS"* and "*beta-tubulin"* is impacted by mutation. The desired output from the relation extraction task is the pair of the normalized Entity Ids, i.e the Gene Entrez IDs of these proteins "<817592, 827837>".

The dataset for this corpus consists of 597 training documents. We analysed 1500 test documents. Limited annotations are provided in the training set. Only positive relations, specifically the NCBI Gene IDs of the interacting protein pairs impacted by mutation, are annotated at the document level. In addition, entity annotations are only provided for those proteins that are referenced in a relation annotation (that is, entity mentions where the entity does not participate in a relevant PPIm relation are not identified). The test set does not have any entity annotations; performance on the test data is therefore inherently limited by the performance of the entity recognition method.

A. Approaches to Relation Extraction

We explored two approaches for relation extraction. Our first approach is a rule-based system wherein a co-occurrence of two protein mentions within the scope of a document is considered to be in a PPIm relation, if it meets certain heuristically designed criteria. In the second approach, we use a binary classifier based on dependency graphs to detect if a given protein pair is in a "PPIm relation" or not.

B. Entity Recognition

Both these approaches require the identification of protein mentions in the text. We used GNormPlus [16] to recognise and normalise protein mentions. GNormPlus had a recall of 53.4%, precision 40.5% and F-score of 46.1% for the protein name normalisation task on the full training set, with the low performance attributable to the fact that not all protein mentions are annotated, as stated above. Applying passage level co-occurrence using the proteins recognised, which considers every possible combination of proteins mentioned within a document to produce protein pairs, has a maximum recall of 39% for relation extraction, based on analysis of the training data. Therefore, even if the relation extraction system had a precision of 100%, the maximum possible F-score would be 56% using GNormPlus for protein name recognition/normalisation task.

C. Filtered Co-occurrence Relation Extraction (Run 3)

The co-occurrence based approach to recognising PPIm relations makes use of a filtering heuristic that considers sentence support, analogous to the notion of support in association rule mining algorithms [18]. The approach applies the idea that if more than *N* sentences contain a given protein pair within a given document, then that pair can be assumed to participate in a PPIm relationship. This heuristic is based on the fact that all the documents in the training set were relevant to PPIm (i.e, all documents were assumed to have passed the Document Triage filter) and hence at least some protein names mentioned in the abstracts can be assumed to be involved in PPIm relationship. We set the sentence support threshold *N* to 2 after experimentation with the training data.

D. Relation Extraction as supervised classification (Runs 1-2)

In this approach, we train a binary classifier with protein-pairs as examples. Following entity recognition, we generate all possible pairs of entity mentions within a document as instances for our classifier. A given protein pair from a document in the training set is considered a positive example if the corresponding NCBI Gene IDs are annotated as being in a PPIm relation in the relation annotation. Otherwise, it is labelled as a negative example. Entity pairs with the same Gene ID (self-relations) are excluded from the set of examples for the classifier. We train the classifier using the Scikit-learn implementation of SVM [15] with the ASM graph kernel [14]. In the test phase, every possible protein pair from test documents is considered as a candidate instance and the binary classifier is used to infer its label. From the examples with a positive prediction, the corresponding NCBI Gene IDs are extracted. The union of all such Gene ID pairs at the document level forms the output of our relation extraction system.

We used the TEES [12] system for preprocessing the text and generating syntactic dependency graphs for sentences. A given pair of entity mentions can be within a single sentence or could be in two different sentences. For an entity pair, whose mentions are within a single sentence, the dependency graph of the corresponding sentence is used as the instance for classification. We refer to these type of relations as sentence level relations. For protein or entity pairs, whose mentions are in two different sentences, we connect the root nodes of the dependency graphs of the two underlying sentences with an artificially introduced root node, to form a single larger graph covering the two sentences. We refer to these relations as non-sentence relations. As depicted in Table IV, most relations in the training set are sentence level relations.

TABLE IV. PPIM RELATION TRAINING DATA CHARACTERISTICS

Dataset	Number of Examples	Number of Positive Examples	Number of Negative Examples
Sentence level relations	6334	1672	4662
Non- sentence level relations	4567	89	4478

Our relation extraction methodology in this task is similar to the one we applied for Chemical-Induced-Disease relation extraction in [13,17], based on the Approximate Subgraph Matching Kernel (ASM) used with an SVM [15] classifier. More details about this approach can be found in [14]; it hinges on evaluating similarity of the subgraph of the dependency graph connecting the entity pairs to those observed in the training data.

Mutation Context:

As the task requires extraction of interacting pairs in the context of a mutation, we attempted to incorporate mutation-context specific features in our classifier. We added a set of binary indicator features for the 30 interaction terms described in section II.B. Additionally, a relevance score is computed for each document and included as a special feature, using the probability of the document being relevant for the Document Triage task, as computed by the classifier that was developed for that task. We did not observe a notable impact on the relation extraction performance by incorporating the mutation context and document relevance features on cross validation test over training data, likely due to the fact that the mutation context is often not incorporated into the same sentence that expresses the protein interaction.

TABLE V. RESULTS ON THE TRAINING SET (5-FOLD CV FOR SVM)

Performance	Precision	Recall	F1
Co-occurrence	30.8%	24.8%	27.4%
SVM with ASM	30.3%	24.7%	27.2%
SVM with ASM + Mutation Context	30.1%	25.3%	27.5%
SVM with ASM + Mutation Context + Document Relevance Feature	29.3%	25.3%	27.2%

E. Results

In Table V we present the results on the training set for relation extraction, based on 5-fold cross validation for SVM. We submitted 3 runs for the official evaluation. The first two runs used the predictions from the SVM classifier with ASM kernel and Mutation Context and Document Relevance Features (both achieving P: 0.3449, R: 0.3287, F1: 0.3366 in the official results). The second run added a Logistic Regression classifier to compute the probability scores for the predictions. The third run was based on the Filtered Co-occurrence system (achieving a slightly higher Precision but lower Recall; P: 0.3890, R: 0.3010, F1: 0.3394 in the official results).

IV. CONCLUSION

In this paper, we present the approaches developed by the University of Melbourne READ-Biomed team for the Precision Medicine Track. We develop machine learning models by incorporating both text mining tools and manually created term lists for the Document Triage task, increasing F1 score by ~10% compared to the baseline results. Further improvements could be to use sentence similarity related features and applying extensive optimization. For the Relation Extraction task, we achieved similar results with a simple co-occurrence based approach and a graph-based classification approach. Incorporating additional features based on mutation context and document relevance did not improve the relation classifier substantially. Further, errors in the earlier entity recognition phase are likely to have substantially limited the relation extraction performance on the test data.

REFERENCES

1. Feldman R, Sanger J, 2007. The text mining handbook: advanced approaches in analyzing unstructured data. Cambridge University Press.

2. Wei CH, Phan L, Feltz J, Maiti R, Hefferon T, Lu Z. 2017. tmVar 2.0: Integrating genomic variant information from literature with dbSNP and ClinVar for precision medicine. Bioinformatics, btx541.

3. Krallinger M, et al. 2011. The Protein-Protein Interaction tasks of BioCreative III: classification/ranking of articles and linking bio-ontology concepts to full text. BMC Bioinformatics, 12(8), p.S3.

4. Dogan RI, Chatr-aryamontri A, Kim S, Wei CH, Peng Y, Comeau D, Lu Z, 2017. BioCreative VI Precision Medicine Track: creating a training corpus for mining protein-protein interactions affected by mutations. BioNLP 2017, pp.171-175.

5. Cohen AM, Bhupatiraju RT, Hersh, WR, 2004. Feature generation, feature selection, classifiers, and conceptual drift for biomedical document triage. TREC.

6. Doughty E et al. 2010. Toward an automatic method for extracting cancer-and other disease-related point mutations from the biomedical literature. Bioinformatics, 27(3), pp.408-415.

7. Kim S, Kwon D, Shin SY, Wilbur WJ, 2011. PIE the search: searching PubMed literature for protein interaction information. Bioinformatics, 28(4), 597-598.

8. Thomas P, Rocktäschel T, Hakenberg J, Lichtblau Y, Leser U, 2016. SETH detects and normalizes genetic variants in text. Bioinformatics, 32(18), 2883-2885.

9. Caporaso JG, Baumgartner WA, Randolph DA, Cohen KB, Hunter L, 2007. MutationFinder: a high-performance system for extracting point mutation mentions from text. Bioinformatics, 23(14), 1862-1865.

10. Verspoor K, Jimeno Yepes A, Cavedon L, McIntosh T, Herten-Crabb A, Thomas Z, Plazzer JP. 2013. Annotating the biomedical literature for the human variome. Database.

11. Jimeno Yepes, A. and Verspoor, K, 2014. Mutation extraction tools can be combined for robust recognition of genetic variants in the literature. F1000Research, 3.

12. Björne, Jari. 2014. Biomedical Event Extraction with Machine Learning. TUCS Dissertations.

13. Panyam NC, Verspoor K, Cohn T, Ramamohanarao K. 2016. Exploiting Tree Kernels for High Performance Chemical Induced Disease Relation Extraction. Proceedings 7th International Symposium on Semantic Mining in Biomedicine.

14. Panyam NC, Verspoor K, Cohn T, Ramamohanarao K. Biomedical Relation Extraction. 2016. ASM Kernel: Graph Kernel using Approximate Subgraph Matching for Relation Extraction. Australasian Language Technology Association Workshop.

15. Pedregosa, F. et al 2011. Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research.

16. Wei CH, et al. 2015. "GNormPlus: an integrative approach for tagging genes, gene families, and protein domains." BioMed Res Int 2015: 918710.

17. Panyam NC, Verspoor K, Cohn T, Ramamohanarao K. 2017. Exploiting Graph Kernels for High Performance Biomedical Relation Extraction. (In Press) Journal of Biomedical Semantics.

18. Agrawal R, Imieliński T, Swami A. 1993. Mining association rules between sets of items in large databases. ACM SIGMOD (22:2, 207-216).

Exploring a Deep Learning Pipeline for the BioCreative VI Precision Medicine Task

Tung Tran[1] and Ramakanth Kavuluru[1,2]
[1]Department of Computer Science, University of Kentucky, Lexington, KY, USA
[2]Division of Biomedical Informatics, Department of Internal Medicine, University of Kentucky, Lexington, KY, USA

***Abstract*—The BioCreative VI Track IV (Mining protein interactions and mutations for precision medicine) challenge is presented to support the Precision Medicine Initiative with the goal of applying biomedical text mining methods to extract interacting protein pairs that are affected by a genetic mutation. Herein, we describe systems that we submitted for the two challenge tasks and report our results. For the document triage task, we report a system performance of 67.23% F1 with a precision of 54.73% and recall of 87.12% on the official test set. For the relation extraction task, our models placed 2nd when using Entrez Gene IDs and 1st when using HomoloGene IDs. When using HomoloGene IDs, our system achieves 37.27% F1 with a precision and recall of 45.44% and 31.61% respectively.**

***Keywords*— deep learning; named entity recognition; relation extraction; protein-protein interactions; gene normalization**

I. Introduction

Precision Medicine (PM) is an emerging disease treatment paradigm in which healthcare is customized to each individual patient. To support this effort, it is important to be able to extract useful translational information such as mentions of relationships between genes, mutations, and diseases. The BioCreative VI Track IV (Mining protein interactions and mutations for precision medicine) challenge is presented in an effort to identify and study mutations and their effect on molecular interactions. This particular track involves two distinct tasks: document triage and relation extraction. In the first task, participants are asked to build systems able to determine whether a PubMed citation is *relevant* or *not relevant*; i.e., whether or not it describes an interaction affected by a genetic mutation. The training set for this first task contains 4082 articles with 1729 of them deemed relevant.

A subset of the aforementioned training documents that are *relevant*, consisting of 597 documents, has been additionally annotated with gene mentions and relevant interaction pairs. This subset is supplied as training data for task 2. In the second task, participants are asked to extract interacting protein pairs that are affected by a mutation. We treat the two tasks independently and built separate systems for each. We describe the system used for the document triage task in Section 2. We then describe the system pipeline used for the relation extraction task in Section 3. Finally, we discuss our validation results in Section 4.

Our work is primarily supported by the National Library of Medicine through grant R21LM012274 and the National Cancer Institute through grant R21CA218231.

II. Document Triage System

We propose using a deep neural network architecture based on convolutional neural networks (CNNs) to tackle the document triage task. The following model is based on the one proposed by Kim et al. (1) for text classification. The input is a document with words $\boldsymbol{w} = (w_1, w_2, .., w_n)$ each represented by their corresponding index to the vocabulary V^{word}. The words are mapped to word vectors via a word embedding matrix $E^{\text{word}} \in \mathbb{R}^{|V^{\text{word}}| \times d}$ to produce a document matrix $D \in \mathbb{R}^{n \times d}$ where d is the length of the word representation vectors. More concisely,

$$D = \begin{pmatrix} E^{\text{word}}[w_1] \\ \vdots \\ E^{\text{word}}[w_n] \end{pmatrix}$$

where $E^{\text{word}}[i]$ is the ith row of E^{word}. The word embedding matrix can be initialized to random or pretrained values; in either case, the word vectors are (further) modified via backward propagation. The central idea in CNNs is the so called *convolution* operation over the document matrix to produce a feature map representation using a convolution filter (CF). The convolution operation $*$ is formally defined as the sum of the elementwise products of two matrices. That is, for two matrices A and B of same dimensions, $\mathrm{A} * \mathrm{B} = \sum_{\mathrm{j}} \sum_k A_{j,k} \cdot B_{j,k}$. With this, a CF is the matrix $W \in \mathbb{R}^{h \times d}$ that is applied as a convolution to a window of size h over D to produce a feature map $v = [v_1, \dots, v_{n-h+1}]$, such that

$$v_i = \text{relu}(W * D_{i:i+h-1} + b)$$

where $D_{i:i+h-1}$ is a window of matrix D spanning from row i to row $i + h - 1$, W and $b \in \mathbb{R}$ are learned parameters, and $\text{relu}(x) = \max(0, x)$ is the linear rectifier activation function. The goal is to learn multiple CF that can collectively capture diverse representations of the same document. Suppose there are k filters, then we produce k corresponding feature maps $v^1, \dots, v^k$. We select the most distinctive feature of each feature map using a max-over-time pooling operation (2) to produce the final feature vector $p \in \mathbb{R}^k$, such that $p = [v^1_{max}, \dots, v^k_{max}]$ where $v^j_{max} = \max(\boldsymbol{v}^j_1, \dots, \boldsymbol{v}^j_{n-h+1})$.

We can also learn different sets of k CFs for different window sizes h as is typically the practice. Choosing a larger h provides more context and thus could be beneficial in improving predictive power but might adversely affect efficiency given the additional time needed. We can then take

the corresponding feature vector for each window size and concatenate them to form the final feature vector. More formally, we can parameterize the window sizes as a sequence $h_1, \dots, h_H$ of H unique sizes. Suppose p^{h_i} denotes the feature vector produced on k filters with a window size of h_i, then the final $kH \times 1$ feature vector is

$$p^* = p^{h_1} \parallel \cdots \parallel p^{h_H}$$

where $\parallel$ is the vector concatenation operation. The output layer consists of m units (one per each of the m target labels) and is fully connected to the full feature vector p^*. The output vector $q \in \mathbb{R}^m$ is thus defined as

$$q = W_q p^* + b_q$$

where $W_q \in \mathbb{R}^{m \times kH}$ is a parameter matrix and $b_q \in \mathbb{R}^m$ is the vector of bias terms. In order to get a categorical distribution, we apply a softmax layer to the vector q such that

$$p_j = \frac{e^{q_j}}{\sum_{j=1}^{m} e^{q_j}}$$

where $p_j \in [0,1]$ is the probability estimate of the label at index j. We optimize on the standard categorical cross-entropy loss via a variant of stochastic gradient descent (SGD) called RMSProp (9). Much of the formulation presented is purposely kept abstract since it will be re-used in a later section. Clearly, the triage task is a binary classification problem where $m = 2$. The configuration of the model implementation is as follows. We used word embeddings of size 200 pre-trained on the PubMed corpus (3). For the convolutional component, we use window sizes of 3, 4, and 5 with 200 convolutional filters. The model is trained for 30 epochs with a mini-batch size of 8 and learning rate of 0.001. Since each instance is a collection of sentences and the window size is at most 5, we pad four zero-vectors at the beginning and the end of the input text as well as between sentences. Moreover, we use a dropout rate of 50%. During training, we checkpoint model parameters at each epoch and only keep the checkpoint resulting in the highest F1 on the development set. We train 10 such models as part of an ensemble. Each model of the ensemble is trained and tuned on a random split of 80% to 20% and seeded with a different value for random parameter initialization.

III. Relation Extraction System

For the second task we propose a pipeline system that consists of three components: supervised named entity recognition (NER) for gene mention detection, knowledge-based gene normalization, and supervised relation classification to predict each pair of genes found as either positive or negative for an interaction. It is possible to use an "out-of-the-box" solution such as GNormPlus (4) that identifies both gene mentions and their corresponding gene identifier directly; however, we opted for a supervised approach that allows us to leverage the generous gene annotations provided with the training corpus for this task. We describe the NER system used to identify potential gene mention spans along with the training corpus in Section 3A. We then describe our system for gene normalization in Section 3B. Lastly, we describe the model we used for relation classification in Section 3C.

A. Gene Mention Identification

The aim of the first component in the pipeline is to identify spans of text that refer to specific genes. To that end, we propose the use of a deep neural network system based on a CNN-LSTM hybrid model proposed by Chiu et al. (5) for NER. This sequence-to-sequence model composes word representations with CNNs by convolving over character n-grams. At the word level, contextual word representations are composed using a bi-directional LSTM layer. A separate fully-connected softmax output layer is present at the output of each LSTM unit such that an Inside, Outside, Beginning (IOB) label prediction can be made for each token.

Herein, we formulate the model from the bottom up while re-using some notations and definitions established in Section 2. In this formulation, a word w_i for $i \in [1, n]$ is treated as a lowercased character sequence $c_1^i, \dots, c_{T^i}^i$ representing their index into the character vocabulary V^{char}. The corresponding character embedding matrix $E^{\text{char}} \in \mathbb{R}^{|V^{char}| \times 32}$ embeds each character as a vector of length 32. We use the same embedding setup to produce character type embedding vectors of length 8 indicating the type of character: lowercase, uppercase, punctuation, or other. This allows the model to generalize specific words while still considering the nuances involved in how named entities presented. Suppose the embedding matrix for *character type* is $E^{\text{ctype}} \in \mathbb{R}^{4 \times 8}$ and $z_1^i, \dots, z_{T^i}^i$ represents the sequence of enumerated character type for the word at position i. The word at position i can then be represented as a matrix composition B^i of its character embeddings, or concretely

$$B^i = \begin{pmatrix} E^{\text{char}}[c_1^i] \parallel E^{\text{ctype}}[z_1^i] \\ \vdots \\ E^{\text{char}}[c_{T^i}^i] \parallel E^{\text{ctype}}[z_{T^i}^i] \end{pmatrix}$$

where $E^{\text{char}}[j]$ and $E^{\text{ctype}}[j]$ are the jth rows of E^{char} and E^{ctype} respectively. We perform a convolution operation over B^i of window size 3 to obtain the feature map $v^i = [v_1^i, \dots, v_{T^i-2}^i]$ such that

$$v_j^i = \text{relu}\left(W^{\text{char}} * B_{j:j+2}^i + b^{\text{char}}\right)$$

where $B_{j:j+2}^i$ is a window of matrix B^i spanning from row j to row $j+2$ and W^{char} and b^{char} are network parameters. We apply 50 filters to obtain 50 corresponding feature maps denoted as $v_1^i, \dots, v_{50}^i$. Let $v_j^i(x)$ be the xth value of v_j^i, then the word representation at position i is $u_i = [\hat{v}_1^i, \dots, \hat{v}_{50}^i]$ where $\hat{v}_j^i = \max\left(v_j^i(1), \dots, v_j^i(T^i - 2)\right)$.

Once a word representation is composed for each word, we can use a bi-directional LSTM to model the sequence. It is important that we also include actual word embeddings as well as *word type* embeddings as input. The latter embeddings serve a similar purpose to that of the character types and can correspond to one of the five following classes: all lowercase, mixed-cased, capitalized, all uppercase, or other. Drawing from

the notation presented in Section 2, the input is a sequence of word indexes $w_1, w_2, .., w_n$ into the word vocabulary V^{word} and the corresponding embedding matrix is denoted as $E^{\text{word}} \in R^{|V^{\text{word}}| \times d}$. In addition, we denote $\bar{z}_1, \bar{z}_2, .., \bar{z}_n$ as a sequence of enumerated *word types* corresponding to the embedding matrix $\mathrm{E}^{\text{wtype}} \in \mathrm{R}^{5 \times 32}$. The bi-directional LSTM can then be composed as

$$\vec{h}^i = \text{LSTM}^{\rightarrow}\left(u_i \parallel E^{\text{word}}[w_i] \parallel E^{\text{wtype}}[\bar{z}_i]\right),$$

$$\overleftarrow{h}^i = \text{LSTM}^{\leftarrow}\left(u_i \parallel E^{\text{word}}[w_i] \parallel E^{\text{wtype}}[\bar{z}_i]\right),$$

$$h^i = \vec{h}^i \parallel \overleftarrow{h}^i \quad \text{for } i = 1, \dots, n$$

where $E^{\text{word}}[j]$ and $E^{\text{wtype}}[j]$ are the jth rows of E^{word} and E^{wtype} respectively. Moreover, $\text{LSTM}^{\rightarrow}/\text{LSTM}^{\leftarrow}$ represent an LSTM unit composition in the forward/backward direction. The output at each timestep necessarily has its own softmax output layer in order to be able to tag each word with an IOB label. The output layer at position i, or $q_i \in \mathbb{R}^m$, is defined as

$$q_i = W_q p_i + b_q$$

where $W_q \in \mathbb{R}^{m \times 50}$, $b_q \in R^m$ are parameters. In order to get a categorical distribution, we apply a softmax layer to the vector q_i such that

$$p_i^j = \frac{e^{q_i^j}}{\sum_{j=1}^{m} e^{q_i^j}}$$

where $p_i^j \in [0,1]$ is the probability estimate of the label at index j for the word at position i. Again, we optimize on the standard categorical cross-entropy loss. However, since each instance may be of a different sequence length, we optimize on the mean loss computed over the n output layers.

Since we are only concerned with gene entities, the label space consists of: B-GENE, I-GENE, and O. These labels are sufficient to indicate whether the token is part of a gene mention. This model is trained on the supplied training data and additionally on the GNormPlus corpus (4) which include re-annotations of the BioCreative II GM/GN corpus (6). The core training data consists of 5668 sentence-level training examples while the GNormPlus corpus constitutes an additional 6389. We used the same pre-trained word embedding vectors of size 200 as described in Section 2 for the document triage task. The network was trained using SGD with an exponential decay rate of 0.95 for a maximum of 10,000 iterations. On each iteration, we trained the network using a mini-batch of 20 random examples. We check-pointed every 100 iterations and saved only the checkpoint with the best F1 on the development set. We also deployed *early stopping* such that training is stopped if there are no improvements for 10 checkpoints. Like in the document triage task, we train 10 such models (each with a different seed) as part of an ensemble where each model is trained on a smaller random subset of only 50% of the original training set.

B. Entrez Gene ID Normalization

For the gene normalization component, we initially experimented with a naive approach using a the gene lexicon provided with the BioCreative II Gene Normalization training data as well as mappings provided with the training corpus. This served as a reasonable baseline; however, it does not take context into consideration during the mapping process. A gene mention may be incorrectly mapped to one of its many homologs resulting in increased false positives. We report the performance of this baseline system in Section 4. The final version of our gene normalization system is knowledge-based and more sophisticated in that it takes into consideration both the gene mention and the context. This system relies on the NCBI gene database [7] to identify the candidate gene IDs for a particular mention and further narrows it down to a best guess based on the document in which it occurred. We define two utility functions that serve as the basis for this system. The first function, $gene_name_lookup$, takes as input a mention span and returns a list of candidate gene IDs sorted by relevance. This is achieved by querying the NCBI gene database via the E-utilities API[1]. This provides a ranked of candidate genes for a given gene mention. The intuition here is that the top few in this list are either the correct gene or at least homologs of the correct gene. We now define the second function, $gene_pmid_lookup$, which takes as input a PMID and returns a list of candidate gene IDs for the article. We achieve this by making another query to the NCBI gene database using the PMID of the current document as query input[2]. This allows us to narrow down the list of candidate gene IDs to ones that have already been identified as appearing in the document. The final gene normalization algorithm takes as input a gene mention and a PMID and returns either a gene ID or *NULL*. The latter indicates that no match can be found, in which case we simply ignore the span entirely for the remainder of the pipeline. The algorithm is defined as follows.

Algorithm 1 Gene Normalization

Input a: gene mention
Input b: document PMID

```
X ← gene_name_lookup(a)
Y ← gene_pmid_lookup(b)
for x ∈ X do
    if x ∈ Y then
        return x
    end if
end for
return NULL
```

For example, suppose the document in question has PMID 18725399 and we wish to map the gene mention "Utp21" to a gene ID. The above algorithm will correctly return 851125 as the gene ID which can be verified via footnotes 1 and 2.

[1] An example query for the gene span "Utp21": https://eutils.ncbi.nlm.nih.gov/entrez/eutils/esearch.fcgi?db=gene&term=Utp21&retmax=100&sort=relevance

[2] An example query for the PMID 18725399: https://eutils.ncbi.nlm.nih.gov/entrez/eutils/esearch.fcgi?db=gene&term=18725399[PMID]

C. Relation Classification of Gene Pairs

For the relation classification component, we use the same CNN architecture introduced in Section 2 modified to suit the problem of relation classification. The model hyper-parameters mirror that of the triage task and the overall task remains binary classification. However, each pair of candidate genes in an article constitutes a separate instance. For each pair of candidate genes, we generate a training instance by performing *entity binding*; i.e., we replace mentions of the pair with GENE_A and GENE_B in the corresponding document text. For a gene pair (A, B), we also generate an additional instance for the reverse case (B, A) given directionality does not matter. Note that we run both cases of a candidate pair during testing and consolidate the predictions. We also generate examples for the exception case when the candidate pairs are the same genes, i.e. $A = B$, in which case GENE_S is used for entity binding of the single gene ID. We also replace mentions of other genes with GENE_N in either case. In total, we generated 2972 instances from the 597 articles in the training set. At test time, we only predict pairs as positive where the mean probability is above 50% for the instance generated from (A, B) and its reverse case (B, A). In case no pairs meet the threshold, we try to make at least one prediction by predicting the pair with the highest probability (even if it is ≤ 50%) as positive for an interaction.

IV. Results and Discussion

For the document triage task, we evaluated performance by training on a held-out set of 70 documents. The system achieved 73.88% F1 with a precision and recall of 67.54% and 81.53% respectively. There may be nuances to this task that is not being effectively captured by the model. On the official test set, the system achieved 67.23% F1 with a precision of 54.73% and recall of 87.12% respectively

TABLE I. System Test Performance

System Evaluation Method	*P (%)*	*R (%)*	*F (%)*
Using Entrez Gene ID	36.53	25.61	30.11
Using HomoloGene IDs	45.44	31.61	37.29

To evaluate the performance of our system pipeline for the relation extraction task, we tested the system on a held-out set of 70 documents. The following are evaluations based on the individual components and are mutually independent. For the NER system, the performance is at 59.48% F1. For the gene normalization component, our initial naive approach using a gene mention lexicon yielded an accuracy of 49.67%. This poor performance may be due to the fact that the lexicon is outdated (2008 release) and that the context of a gene mention is generally ignored with this approach. The more sophisticated version that operates by cross-referencing the NCBI gene database yielded a much improved accuracy of 78.90%. The final component in the pipeline responsible for relation classification yielded an F1 of 81.80% with a precision of 72.57% and recall of 93.71%.

We now report *end-to-end* performance of the system for relation extraction on the held-out validation set. If we assume perfect gene annotations, we can extract pairs with reasonable performance at an F1 of 83.72%. NER and gene normalization represent a major barrier to achieving such promising results overall. Our initial experiments show that with naive gene normalization, the system performed with an F1 score of 20.62% with a precision and recall of 20.41% and 20.83% respectively. The low performance here is expected since incorrect gene annotations will propagate to the end of the pipeline; nevertheless, it serves as a baseline for comparison. The PubTator tool (8), which uses GNormPlus as the backend for gene annotations, results in more than double the recall at 43.72% but still suffers from low precision at 19.27% with an overall F1 of 26.75%. If we use the more sophisticated gene normalization method from Section 3.2, we observe improved precision and recall at 28.82% and 51.04% respectively and an overall performance of 36.84% F1 on the validation set.

The performance of our system on the official test set is recorded in Table 1. When evaluating on original Entrez Gene IDs, we achieved a system performance of 30.11% F1 with a precision and recall of 36.53% and 25.61% respectively which ranked second among systems that were submitted. When evaluating on HomoloGene IDs, which accounts for homologous genes, we achieved a system performance of 37.27% F1 with a precision and recall of 45.44% and 31.61% respectively which ranked first among submitted runs.

References

1. Kim, Y. (2014) Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Association for Computational Linguistics, 1746–1751.

2. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011) Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, vol. 12, 2493–2537.

3. Moen, S. and S. S. Ananiadou, T. (2013) Distributional semantics resources for biomedical text processing. In *International Symposium on Languages in Biology and Medicine (LBM)*, 39-44.

4. Wei, C.-H., Kao, H.-Y., and Lu, Z. (2015) GNormPlus: an integrative approach for tagging genes, gene families, and protein domains. *BioMed research international*, vol. 2015.

5. Chiu, J. P. and Nichols, E. (2016) Named entity recognition with bidirectional lstm-cnns. *Transactions of the Association for Computational Linguistics*, vol. 4, 357–370.

6. Morgan, A. A., Lu, Z., Wang, X., Cohen, A. M., Fluck, J., Ruch, P., Divoli, A., Fundel, K., Leaman, R., Hakenberg, J., et al. (2008) Overview of biocreative ii gene normalization. *Genome biology*, vol. 9, no. 2, p. S3.

7. Maglott , D., Ostell , J., Pruitt , K. D., and Tatusova, T. (2010) Entrez gene: gene-centered information at ncbi. *Nucleic acids research*, vol. 39, no. suppl 1, D52–D57.

8. Wei, C.-H., Kao, H.-Y., and Lu, Z. (2013) PubTator: a web-based text mining tool for assisting biocuration. *Nucleic Acids Res*, 41(W1):W518-22.

9. Tieleman, T. and Hinton, G. (2012) Lecture 6.5---RmsProp: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning

Text Classification using Ontology and Semantic Values of Terms for Mining Protein Interactions and Mutations

Berna Altınel [1], Zehra Melce Hüsünbeyi [2] and Arzucan Özgür [3]

[1] Department of Computer Engineering, Marmara University, TR-34722, Kadıköy, Istanbul, Turkey

{berna.altinel}@marmara.edu.tr

[2,3] Department of Computer Engineering, Boğaziçi University, TR-34342, Bebek, Istanbul, Turkey

{arzucan.ozgur,zehra.husunbeyi}@boun.edu.tr

Abstract*—There is a big amount of published articles in the biomedical domain. Automatically processing and extracting biologically crucial information such as protein-protein interactions from the biomedical literature is one of the main difficulties for both research and commercial platforms. We contributed to the Document Triage Task of the Precision Medicine Track of the BioCreative VI challenge assessment by developing text mining methods to assist health professionals and researchers. We build three methods, which are capable of receiving a list of PMIDs and returning a relevance-ranked judgement for the articles in order to identify relevant PubMed articles, which are about genetic mutations affecting protein-protein interactions. Our first approach is based on meaning calculation, which computes the words' meaning scores in the scope of classes. Meaning computation is based on the Helmholtz principle and has been utilized for various applications in the field of text mining like feature extraction, information retrieval, text classification, and text summarization. Nevertheless, to the best of our knowledge, our effort is the first work, which uses meaning calculation of the terms retrieved from the Interaction Network Ontology (INO) to construct a semantic classifier. Our second approach depends on the extraction of the most salient terms, generated from Genia Tagger[1], taking advantages of the term frequency-relevance frequency (TF-RF) metric. This methodology also uses sprinkling, which is a process of adding further terms corresponding to class labels of documents to the training documents in order to strengthen class-based relationships in the training phase. Our third approach is based on using a Convolutional Neural Network (CNN). In our model, the first layer embeds words into low-dimensional vectors. Convolutions over the embedded word vectors are performed in the next layer by using multiple filter sizes. The next stage is max-pooling the result of the convolutional layer into a long feature vector. Finally, the results are classified with the softmax layer. In order to evaluate our results, we also implemented linear kernel with Support Vector Machines (SVM) and Naïve Bayes (NB) as the baseline algorithms. According to the experiment results the presented methods outperform the baseline algorithms.

Keywords*— *text mining, meaning, TF-RF, sprinkling, CNN.

[1] http://www.nactem.ac.uk/GENIA/tagger/

I. Introduction

The huge volume of published articles in the scientific literature continues to enlarge by the contributions of millions of people every day. It is vitally important to extract clinically useful information that links genes, mutations, and diseases to specialized treatments from this published literature for the precision medicine initiative (PMI), which aims to find individualized treatment for a patient according to his/her genetic profile.

Protein-protein interactions (PPIs) are crucial for a range of biological processes such as cell cycle control, DNA replication, signal transduction etc. Mutations may affect the stability and affinity of protein-protein interactions. Thus, combining the efforts in protein-protein interaction [1, 2] and mutation extraction [3] has high significance for precision medicine.

Donaldson et al. [4] present two classifiers for finding protein-protein interaction data in PubMed. According to their experiments performed on the BIND database [5], the SVM method with linear kernel is reported to give higher classification results than a Naïve-Bayesian classifier. Mitsumori et al. [6] present one of the other systems where SVM is used. They use the bag-of-words (BOW) feature representation for the words that are closest to the protein names in order to get protein-protein interactions. They compare their own system's capability on extracting protein-protein interactions, to that of other systems' in the literature. A semi-supervised information extraction approach for identifying sentences in text that show an interaction relation between two proteins is suggested in [7]. This methodology depends on the analysis of the paths between two protein names in the dependency parse trees of the sentences. They mention that they get significant improvement over the results of the existing methods in the literature. Participants in the Protein-Protein Interaction tasks of the BioCreative II and BioCreative III Challenges also attempt to develop methods that aim to detect interaction relevant articles by using numerous techniques. Alex et al. [8] apply a SVM classifier

with the usage of pre-processing, part-of-speech (POS) tagging, sentence splitting and shallow parsing. They report 70% precision, 86% recall and 77% F-score. Another study present a methodology which integrates protein name detection and abbreviation resolution systems by also using SVM and obtain 78% F- score [9].

In this paper, we present our participation in the Document Triage Task of PrecMed Track of the BioCreative VI Challenge assessment by developing text-mining methods to assist health professionals and researchers. We present three methods for identifying the articles about genetic mutations affecting protein-protein interactions. Our approaches are based on class-based semantic values of terms, ontology and convolutional neural networks.

II. RELATED WORK

Term Frequency-Relevance Frequency (TF-RF)

Term Frequency-Relative Frequency (TF-RF) is presented by Lan et al. [9] and is a supervised term weighting method, which is based on the number of positive and negative documents that a term occurs in. In a text classification setting a selected category is labeled as the positive category, while all other categories in the same dataset are labeled as the negative category. The formula of TF-RF is:

$$TF-RF = tf_w \times \log\left(2 + \frac{a}{\max(1,c)}\right) \qquad (1)$$

where tf_w shows the term frequency of word w, a denotes the number of documents in the positive category which include word w, and c is the number of documents in the negative category which include word w. According to an explanatory example given in [9]; in contrast to IDF, with RF methodology each word is assigned more appropriate weights from the point of different categories since RF considers the category information.

Helmholtz Principle from Gestalt Theory and Meaning Calculation

According to the Helmholtz principle in human perception from Gestalt Theory, humans easily notice events with a large deviation from noise or randomness [10]. A number of explanatory examples are given by Balinsky et al. [10] and these examples show that interesting events and meaningful features appear in great deviations from randomness.

Textual data comprise structures like sentences, paragraphs and documents. Balinsky et al. [10] try to describe the meaningfulness of these structures by utilizing the Helmholtz principle. A meaning value is given to each word for modeling the meaningfulness of these structures. Balinsky et al. [10] mention that a sharp rise in frequencies can be used in quick modification discovery. A burst is a period of increased and quick modifications in an event as mentioned in [11].

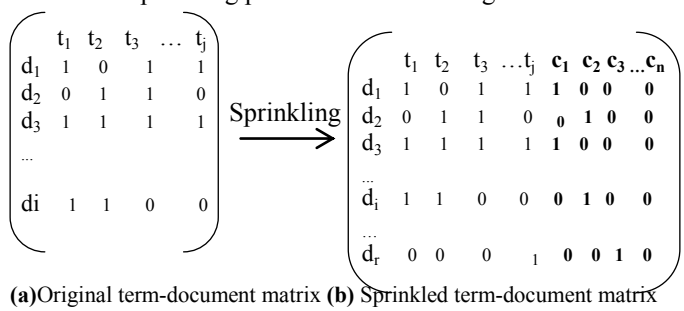
The meaning value of a term (word) w in a class c_j is computed with Eq. (2) [10]:

$$meaning(w, c_j) = -\frac{1}{m}\log\binom{K}{m} - \left[(m-1)\log N\right] \qquad (2)$$

where w denotes a word, m shows the occurrence of term w in class c_j, K specifies the frequency of term w in the whole dataset. $N=L/B$; L represents the length of the dataset and B represents the length of the class c_j in number of terms [12]. If a word's meaning score in a specific class is larger, then this means that this word is more informative for that class. A meaning value of a word essentially shows how high this word's frequency is likely to be in a class of documents compared to the other classes of documents.

Sprinkling

Sprinkling is a process of adding further terms corresponding to class labels of documents to the training documents in order to strengthen class-based relationships in the training phase. For instance, in [13] Latent Semantic Indexing (LSI) is performed both on standard term-document matrix and term-documents matrix augmented with sprinkled terms. The sprinkling process is shown in Figure 1:



(a) Original term-document matrix **(b)** Sprinkled term-document matrix

Figure 1. (a) Original term-document matrix with r documents and j terms. **(b)** term-document matrix after sprinkling with n terms. These new additional terms show the class labels of the corresponding documents. For instance, d_1 belongs to class c_1, d_2 belongs to class c_2, d_3 belongs to class c_1, d_i belongs to class c_2…. etc.

Chakraborti et al. [13] drop sprinkling terms after performing LSI. Test documents are classified using k-Nearest Neighbors (kNN) with the Euclidean distance metric. According to their experimental results the presence of sprinkling terms improves the classification performance. For instance, the classification accuracies of Sprinkled-LSI on four different subgroups of 20NewsGroups[1] dataset are reported as 86.99%, 80.60%, 80.42%, and 93.89% while the classification accuracies of LSI are reported as 79.32%, 72.55%, 66.30% and 91.17%; respectively [13]. They state that the integration of further knowledge, which represents the latent class structure, improves the classification performance.

[1] http://qwone.com/~jason/20Newsgroups/

Consequently; with sprinkling process, documents related to the same class are located closer to each other.

Convolutional Neural Network (CNN)

Convolutional neural networks apply layers with convolving filters, which are used as local features [17]. CNN models have achieved a notable performance in the natural language processing field, particularly in the task of sentence classification [14, 15, 16).

A CNN architecture with multiple convolution layers was proposed by Kalchbrenner [15]. This model consists of positing latent, dense and low-dimensional word vectors (initialized to random values) as inputs.

Kim's [14] model, shown in figure 2, proved that CNN can obtain state-of-the-art results with a simple one layer architecture. This model uses pre-trained word vectors as inputs. This is followed by a convolution and maxpooling layer, and a softmax classifier.

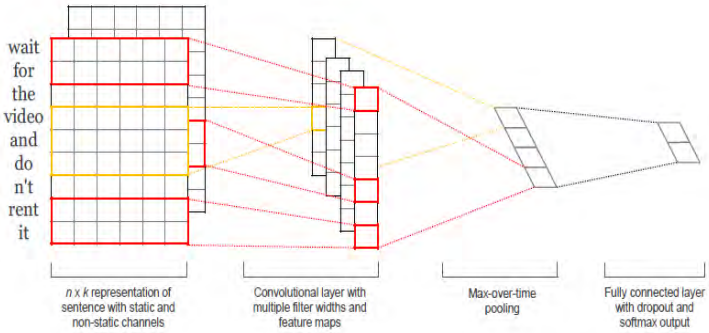


Figure 2. Model architecture with two channels for an example sentence [14].

A similar model is proposed by Johnson and Zhang [16], but they used high dimensional 'one-hot' vector representations of words as CNN inputs. Their focus was on classification of longer texts, rather than sentences.

III. METHODS

A. Semantic Meaning Classifier with INO (SMC-INO)

In a novel text classification algorithm named Supervised Meaning Classifier (SMC) [18], the authors calculate the meaning scores of the words in a document for a certain category and sum them to obtain a relative class membership value of the document for that category. In other words, the class membership of a particular document is determined by the sum of the meaning or the importance of its terms for that particular class. This is somewhat similar to the Naive Bayes algorithm where the class conditional document probability $P(D|C)$ is calculated by multiplying probabilities of the class conditional term probabilities $P(w|C)$ in addition to a class prior probability $P(C)$. The SMC classifier uses meaning calculations as explained in Section II. Ganiz et al. [18] calculate the meaning scores of each word in the training set for each class, which constitutes the training phase. In the classification phase, for an unlabeled new test instance, meaning scores of the words for a particular class are summed up to obtain class membership value. The class with largest membership value is chosen as the label of the instance. SMC is show to be superior to Multinomial NB and SVM with linear kernel, especially on inadequate training data [18].

In our experiment setting, we first find interaction terms by using the Interaction Network Ontology (INO) [19] and generate training and test datasets according to these terms. Then, we use the SMC algorithm for classifying the documents.

B. Sprinkled Relevance Value Classifier (S-RVC)

We implemented a Relevance Value Classifier similar to SMC with the only difference that the class-based term values are calculated by using the TF-RF metric described in Section II. We developed Sprinkled Relevance Value Classifier (S-RVC) which has the same architecture with RVC with the only difference that; S-RVC uses additional terms which represent the class relationships between documents. In other words, class labels are added into the standard term-document matrix in order to enrich the class knowledge in the training corpus and add this information into the classification model.

C. Convolutional Neural Network (CNN)

The CNN model architecture, which is related to Kim Yoon's model [14], includes several layers such as embedding, convolution, max-pooling, dropout and softmax. In the first layer that defines the embedding layer, vocabulary word indices correspond to low-dimensional vectors and embeddings are learned from scratch. Then convolutions are performed on the embedding matrix via linear filters of different sizes (3, 4, 5). Each filter operates for generating a feature map for a window of words. This is followed by a max-pooling operation [16], which is applied, to each feature mapping for inducing a particular feature vector. For regularization, the dropout method is used which avoids co-adaptations of hidden units to decrease overfitting. Finally, with the softmax layer normalized probabilities over labels are generated. In our experiments, the hyperparameters, which are the number and sizes of convolutional filters, dropout rate, and mini-batch size, are chosen with investigating their impacts on performance during the training phase.

IV. EXPERIMENT SETUP

A. Data set

The training dataset of Biocreative VI Document Triage Task has 4082 PubMed articles. These articles are manually labelled as relevant (i.e. describing genetic mutations affecting protein-protein interactions) or not relevant by BioGRID database curators. This dataset has skewed class distribution, since 1729 of these documents are labelled as relevant while 2353 of them are labelled as not relevant. There are 17765 words in the training dataset of Biocreative VI Document Triage Task. The provided training set is further split into training (70%) and test (30%) sets for evaluation purposes as shown in Table 1.

Table 1. Overview of training and test splits of the dataset

Dataset	# Relevant citations	#Not Relevant citations	#total citations
Training	1210	1769	2979
Test	519	584	1103

Interaction Network Ontology (INO) and Genia Tagger

The Interaction Network Ontology [19] was used to select protein related interaction keywords from the literature. We obtained a sequence of literature mining keywords common in the training dataset, which includes 437 tokens. Named entity tags, which are cell lines, cell types, DNA, RNA and protein names, are extracted from the training dataset with the Genia Tagger [20]. In our experiment settings, we used the detected terms via INO and Genia Tagger to increase the performance of the document classification algorithms.

B. Experiment Setting and Evaluation

We apply stemming and stopword filtering to these datasets. After running the algorithms on 10 random training and test splits, we report the average of these 10 results. The evaluation metrics in our experiments are precision/recall and F-Measure.

C. Baseline Algorithms

We use two baseline algorithms in order to evaluate the results of our methods. The first baseline algorithm is the customary linear kernel of SVM. The secondly baseline algorithm is Naïve Bayes.

V. RESULTS AND DISCUSSION

We evaluate and compare the three proposed methods with the two baseline algorithms, which are commonly used high accuracy techniques in the text classification field. The experiment results are shown in Table 2. According to Table 2, the F-scores of the baseline algorithms are 60.3% and 61.4%, for SVM with linear kernel and Naive Bayes, respectively. It is important to note that linear kernel is the traditional state of the art algorithm in SVM for text classificationin domain [21,22]. We achieve our best F-score performance of 88.6% with CNN, which outperforms the two baseline algorithms with a significant difference. While SVM with linear kernel achieves the highest precision of 92.0%, it performs remarkably worse than all other algorithms in terms of F-score. Furhermore, our other two algorithms, S-RVC, SMC-INO, obtain 75.9% and 86.2% F-scores as shown in Table 2. The algorithm which gets the lowest F-score among our algorithms is S-RVC; still it obtains higher F-score in comparison to both linear kernel and Naive Bayes. The superioty of S-RVC over both baseline algorithms could be explained with the usage of class-based relavence values of terms and additional sprinkled terms. Similarly, the superiority of SMC-INO over both baseline algorithms could be explained with the usage of class-based meaning values with the integration of terms retrieved from INO. The advantages of semantic text classification over the traditional text classification are analyzed and discussed in several studies in the literature such as [23, 24, 25, 26]. On the other hand, we get the highest F-score performance over the baseline algorithms with CNN. This may not be surprising, since the high performance of deep learning algorithms in comparison to other existing algorithms are presented in many recent studies [27, 28, 29; 30].

Table 2. Experiment results

Method	*Precision*	*Recall*	*F-Measure*
linear kernel	0.920	0.437	0.603
Naive Bayes	0.832	0.461	0.614
SMC-INO	0.774	0.921	**0.862**
S-RVC	0.755	0.745	**0.759**
CNN	0.863	0.912	**0.886**

Furthermore, SMC-INO, S-RVC and CNN obtain the following precision, recall and F1 values on the real test dataset of the Document Triage Task of the Precision Medicine Track of the BioCreative VI challenge: SMC-INO obtain 0.4886 average precision, 0.5849 recall and 0.5268 F1; S-RVC obtain 0.5055 average precision, 0.7178 recall and 0.5865 F1;CNN obtain 0.5098 average precision, 0.9795 recall and 0.6685 F1.

VI. CONCLUSION AND FUTURE DIRECTIONS

We contributed to the Document Triage Task of PrecMed Track of the BioCreative VI challenge assessment by presenting three methods identifying PubMed articles which are relevant to genetic mutations affecting protein-protein interactions. We also implement two baseline algorithms. We achieve our best F-score performance of 88.6% with CNN, which outperforms the two baseline algorithms with a large margin. Our experimental results show the promise of our novel techniques, S-RVC and SMC-INO. To the best of our knowledge, our's is the first attempt to build these approaches and apply them in the BioNLP domain. Moreover, CNN shows remarkable superiority over the baseline algorithms and it forms a foundation that is open to several improvements. As future work, we would like to analyze and shed light on how our new approaches and CNN implicitly capture semantic information in the context of a class when calculating the similarity between two documents. We also would like to develop a semantic kernel and compare it with the traditional linear kernel. In addition, we plan to implement an algorithm for the relation extraction Task of the PrecMed Track of the BioCreative VI challenge assessment and further improve our approaches.

ACKNOWLEDGMENT

We would like to thank the BioCreative Task organizers for organizing the shared task and for their help with the data preparation and the questions.

REFERENCES

1.Kim S, Islamaj Dogan R, Chatr-Aryamontri A, Chang CS, Oughtred R, Rust J, et al.(2016). BioCreative V BioC track overview: collaborative biocurator assistant task for BioGRID. Database : the journal of biological databases and curation.

2.Krallinger M, Vazquez M, Leitner F, Salgado D, Chatr-Aryamontri A, Winter A, et al. The Protein-Protein Interaction tasks of BioCreative III: classification/ranking of articles and linking bio-ontology concepts to full text. BMC bioinformatics. 2011;12 Suppl 8:S3.

3. Wei CH, Harris BR, Kao HY, Lu Z. tmVar: a text mining approach for extracting sequence variants in biomedical literature. Bioinformatics. 2013;29(11):1433-9.

4. Donaldson, J. Martin, B. de Bruijn, C. Wolting, V. Lay, B. Tuekam, S. Zhang, B. Baskin, G. D. Bader, K. Michalockova, T. Pawson, and C. W. V. Hogue. (2003) Prebind and textomy - mining the biomedical literature for protein-protein interactions using a support vector machine. BMC Bioinformatics, 4:11.

5. Bader, G. D., Betel, D., & Hogue, C. W. (2003). BIND: the biomolecular interaction network database. Nucleic acids research, 31(1), 248-250.

6. Mitsumori, T., Murata, M., Fukuda, Y., Doi, K., & Doi, H. (2006). Extracting protein-protein interaction information from biomedical text with SVM. IEICE Transactions on Information and Systems, 89(8), 2464-2466.

7. Erkan, G., Özgür, A., & Radev, D. R. (2007, June). Semi-supervised classification for extracting protein interaction sentences using dependency parsing. In EMNLP-CoNLL (Vol. 7, pp. 228-237).

8. Alex, B., Grover, C., Haddow, B., Kabadjov, M., Klein, E., Matthews, M., ... & Wang, X. (2008). Automating curation using a natural language processing pipeline. Genome Biology, 9(2), S10.

9. Lan M, Tan CL, Su J, Lu Y. Supervised and traditional term weighting methods for automatic text categorization. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2009;31(4):721-735.

10. Balinsky, A., Balinsky, H., Simske, S., 2011a. On the Helmholtz Principle for Data Mining. In: Proceedings of Conference on Knowledge Discovery, Chengdu, China.

11. Kleinberg, J., 2002. Bursty and Hierarchical Structure in Streams, Proceedings of the 8th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), 7(4):373-397.

12. Balinsky, A., Balinsky, H., Simske, S., 2011b. Rapid change Detection and Text Mining. In: Proceedings of 2nd Conference on Mathematics in Defense (IMA), Defense Academy, UK.

13. Chakraborti, S., Lothian, R., Wiratunga, N., & Watt, S., 2006. Sprinkling: supervised latent semantic indexing. In European Conference on Information Retrieval (pp. 510-514). Springer Berlin Heidelberg.

14. Kim, Y., 2014. Convolutional Neural Networks for Sentence Classification, CoRR.

15. Kalchbrenner, N., Grefenstette, E., & Blunsom, P. (2014). A convolutional neural network for modelling sentences. arXiv preprint arXiv:1404.2188.

16. Johnson, R., & Zhang, T. (2014). Effective use of word order for text categorization with convolutional neural networks. arXiv preprint arXiv:1412.1058.

17. Y. LeCun, L. Bottou, Y. Bengio, P. Haffner. 1998. Gradient-based learning applied to document recognition. In Proceedings of the IEEE, 86(11):2278–2324, November.

18. Ganiz, M. C., Tutkan, M., & Akyokus, S., 2015. A novel classifier based on meaning for text classification. In Innovations in Intelligent SysTems and Applications (INISTA), 2015 International Symposium on (pp. 1-5). IEEE.

19. Özgür, A., Hur, J., & He, Y. (2016). The Interaction Network Ontology-supported modeling and mining of complex interactions represented with multiple keywords in biomedical literature. BioData mining, 9(1), 41.

20. Tsuruoka, Y., Tateishi, Y., Kim, J. D., Ohta, T., McNaught, J., Ananiadou, S., & Tsujii, J. I. (2005, November). Developing a robust part-of-speech tagger for biomedical text. In Panhellenic Conference on Informatics (pp. 382-392). Springer, Berlin, Heidelberg.

21. Joachims, T., 1998. Text Categorization With Support Vector Machines: Learning With Many Relevant Features, pp.137-142, Springer Berlin Heidelberg.

22. Dumais, S., Platt, J., Heckerman, D., Sahami, M., 1998. Inductive Learning Algorithms And Representations For Text Categorization. In: Proceedings of the Seventh International Conference on Information Retrieval and Knowledge Management (ACM-CIKM), pp. 148–155.

23. Bloehdorn, S., Moschitti, A., 2007. Combined Syntactic and Semantic Kernels for Text Classification, Springer, 307-318.

24. Siolas, G., d'Alché-Buc, F., 2000. Support Vector Machines Based On a Semantic Kernel for Text Categorization, Proceedings of the International Joint Conference on Neural Networks (IJCNN), IEEE, 5(1):205-209.

25. Wang, P., Domeniconi, C., 2008. Building Semantic Kernels for Text Classification Using Wikipedia, Proceeding of the 14th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), 713-721.

26. Kim, K., Chung, B. S., Choi, Y., Lee, S., Jung, J. Y., & Park, J., 2014. Language independent semantic kernels for short-text classification. Expert Systems with Applications, 41(2), 735-743.

27. Cao, Z., Li, S., Liu, Y., Li, W., & Ji, H., 2015. A Novel Neural Topic Model and Its Supervised Extension. In AAAI (pp. 2210-2216).

28. Hill, F., and Korhonen, A., 2014. Learning abstract concept embeddings from multi-modal data: Since you probably can't see what I mean. In Proceedings of EMNLP, 255–265.

29. Hinton, G., Salakhutdinov, R., 2011. Discovering binary codes for documents by learning deep generative models. Topics Cogn Sci 3(1):74–91.

30. Le, Q. V., & Mikolov, T., 2014. Distributed representations of sentences and documents. arXiv preprint arXiv:1405.4053.

Leveraging Prior Knowledge for Protein-Protein Interaction Extraction with Memory Network

Huiwei Zhou, Zhuang Liu, Shixian Ning, Yunlong Yang
School of Computer Science and Technology, Dalian University of Technology, Dalian, People's Republic of China

***Abstract*—Extracting Protein-Protein Interaction (PPI) from biomedical literature is an essential step towards precision medicine. This paper proposes a novel PPI extraction approach, which leverages prior knowledge about protein-protein pair with memory network. The proposed memory network-based model (MNM) captures important context clues related to knowledge representations learned from Knowledge Bases (KBs). We perform 5-fold cross validation on the BioCreative VI PPI training set and achieve a *F*1-*score* of 59.67%, which demonstrates the effectiveness of memory network-based approach for the PPI extraction.**

Keywords—protein-protein interaction; relation extraction; memory network; prior knowledge

I. INTRODUCTION

With the rapidly growing of biomedical literature, it is becoming urgent and significant for natural language processing (NLP) experts to develop entity relation automatic extraction technique. The BioCreative VI proposes a challenge task of automated PPI extraction from the biomedical literature, which aims to extract interaction relations between protein entities mentioned within a document. Traditional feature-based relation classification methods [1-4] apply machine learning technique to learn models with the one-hot represented lexical and syntactic features. However, these methods need extensive feature engineering and complicate linguistic analysis, which suffer from time consuming and may lead to error propagation. Recently, deep learning techniques have achieved great success in relation extraction tasks. Without the effort of feature engineering, deep neural networks could effectively extract semantic information for relation extraction. Zeng et al. [5] first employ Convolutional Neural Networks (CNN) [6] to capture the lexical and position information for relation classification, and get better performance than the traditional feature-based classifiers. Peng et al. [7] apply CNN for PPI extraction. Except the embeddings vector of each word in the sentence, they also add the embeddings of the head of each word. Gu et al. [8] adopt CNN to integrate context information and dependency information for chemical-disease relation (CDR) extraction.

Apart from CNN, Recurrent Neural Network (RNN) [9] is another widely used deep neural networks, which shows competitive in relation classification. Zhang and Wang [10] employ RNN to tackle the problem of long-distance dependency between entity pairs. Zhou et al. [11] combine a long short-term memory (LSTM) [12] network with a kernel-based model to capture semantic information and syntactic information. Zhou et al. [13] use bidirectional LSTM (Bi-LSTM) for relation classification. Moreover, an attention mechanism is applied on Bi-LSTM to merge hidden representations from each time step into a sentence-level vector.

All methods mentioned above use texts as resources. Nevertheless, biomedical experts have built many large-scale KBs, which contains PPI triples of protein entity pairs and their interaction relation, such as IntAct [14], BioGrid [15], UniProt [16]. These prior knowledge are crucial resources for PPI extraction.

Recently, neural-based representation learning (RL) methods are also applied to encode relational knowledge with low-dimensional embeddings of entities and relations. Many methods have been proposed for knowledge representations (KRs) learning, among which translation-based models [17-19] are simple and effective with the state-of-the-art performance. TransE [17] is a typical translation-based method, which regards a relation r as a translation from the head entity h to the tail entity t with the $\mathbf{h}+\mathbf{r}\approx\mathbf{t}$ in the embedding space, if the triple (h,r,t) holds.

This paper employs TransE to learn embeddings of protein entities and relations from KBs, which are then introduced to a memory network [20-21] to combined with context embeddings. Experiments on the BioCreative VI PPI dataset show that our approach could be effectively leverage prior knowledge to improve PPI extraction performance. Our approach does not rely on feature engineering and is computationally efficient.

II. METHODS

A. Preprocessing

Each protein pair in a given document is regarded as a candidate instance except that the two proteins occur over more than one sentence. Our model takes word embeddings of sequences between protein pairs and knowledge representations learned from KBs as input to calculate the probability of interaction relation. The numbers in the context are replaced by a specific string, such as "NUMBER". And some special characters, such as "*", are removed. Additionally, we replace all the entity mentions as "gene0" except the pair entities need to classify. Finally, based on the positive statistics of training data, we select the protein pairs, whose distance is more than 5 and less than 50 characters, to be

the final instances. We use Word2Vec tool[1] [22] to pre-train word embeddings on the datasets (about 9,308MB) downloaded from PubMed[2].

B. Knowledge representation learning

We employ TransE model [17] to learn knowledge representations based on the entity-relation triples in protein KBs IntAct [14], BioGrid [15] and UniProt [16].

Note that entity embeddings are initialized with the averaged embeddings of words contained in entity mention and relation embeddings are initialized randomly. We introduce such prior knowledge, embeddings of entity and relation, into the memory network to improve the extraction performance.

C. Architechture of the approach

Memory network is a recurrent attention model over a possibly large external memory which allows to read and write to [20]. To leverage entity embeddings of protein pair, two memory networks are applied to pay attention to the two entity embeddings respectively to select important context words as shown in Fig 1. The two memory networks share the same set of parameters to learn the weights of the context words between the two entities. In addition, the attention mechanism in the two memory networks also share the same parameters, which could enable the two entities to communicate with each other, just like using same eyes to watch two things that may be relevant.

In each memory network, there are two computational layers, each of which contains an attention layer and a linear layer. The outputs of the two networks are concatenated with relation embeddings learned from KBs for relation classification.

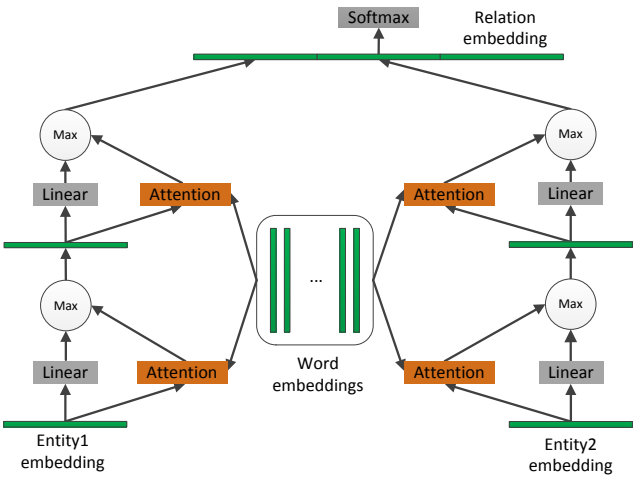


Fig. 1. The architecture of the proposed memory network-based model (MNM). Entity1, Entity2 and Relation representtaions are learned by TransE. Our approach consists of two memory networks, each of which contains two attention layers which share the same parameters. Note that the two memory networks also share the common parameters, namely, that the same attention operation is applied to both Entity1 and Entity2. Finally, the two output vectors of the two memory networks and the relation embeddings are concatenated and sent to the softmax layer for relation classification.

D. Memory network model (MNM)

a) Attention mechanism: Given a word sequence between a pair of protein entities in a document $s = \{w_1, w_2, ..., w_i, ..., w_n\}$, the corresponding word embeddings $\{e_1, e_2, ..., e_i ..., e_n\}$ is regard as the memory $m \in \mathbb{R}^{d \times n}$, where $e_i \in \mathbb{R}^d$ is a *d*-dimensional word vector. As for each attention layer, one entity embedding is concatenated to each piece of memory m_i to select important evidences as follows:

$$g_i = \tanh(W_a[m_i; e_{entity}] + b_a) \tag{1}$$

where $[m_i; e_{entity}]$ denotes the concatenation of $m_i \in \mathbb{R}^{d \times 1}$ and protein embedding $e_{entity} \in \mathbb{R}^{d \times 1}$, $W_a \in \mathbb{R}^{1 \times 2d}$, and $b_a \in \mathbb{R}^{1 \times 1}$. After obtaining $\{g_1, g_2, ..., g_n\}$, the attention weight of each word can be defined as follows:

$$\alpha_i = \frac{\exp(g_i)}{\sum_{j=1}^{n} \exp(g_j)} \tag{2}$$

The attention layer output $output_a \in \mathbb{R}^{d \times 1}$ is then calculated as a weighted sum of each piece of memory in m:

$$output_a = \sum_{i=1}^{n} \alpha_i m_i \tag{3}$$

b) Dimension-wise max pooling: To take an optimal over the context embeddings and entity embedding, we do a dimension-wise max pooling on the attention layer output $output_a$ and the linear transformation of entity embedding e_{entity}. And the max pooling vector is considered as the new entity embedding e_{entity} for the next layer. For the other entity, the same operations are applied. Finally, we concatenate the two max pooling vectors of the second computational layer and the relation embeddings learned from KBs, and fed it to a softmax layer to perform relation classification.

c) Location impaction: Following Sukhbaatar [20], we control the input percentage of each piece of memory by its relative distance to the entity mention. Each percentage is calculated as follows:

$$per_{i,k} = (1 - p_i / n) - (k / d)(1 - 2 \times p_i / n) \tag{4}$$

1 https://code.google.com/p/word2vec/

2 http://www.ncbi.nlm.nih.gov/pubmed/

Where n is sequence length, k is the number of current layer, p_i is the relative distance between current word and entity mention, d is the dimension of word embeddings. Therefore, the actual memory vector is computed with $m_{i,k} = e_i \odot per_{i,k}$.

III. Results and Discussion

We use the BioCreative VI PPI training dataset which has annotated some protein pairs that have relation, and perform a 5-fold cross validation on it. Protein entity recognition and normalization are accomplished with GNormPlus [23] toolkits. The dimension of word, entity, and relation embedding are all 100. The memory network model is trained by using Adam technique [24] with a learning rate 0.001. The whole framework is developed by PyTorch.

A. Effects of prior knowledge

In the experiments, the proposed MNM is compared with the following baseline methods:

AE (Averaged Entity Embeddings): This method represents entities as an average of their constituting word embeddings, but does not concatenate relation embeddings to the outputs of memory networks. Only entity embeddings are employed. And relation embeddings are not used.

TE (TransE-based Entity Embeddings): This method employs TransE-based entity embeddings learned from KBs, but does not concatenate relation embeddings to the outputs of memory networks. Only entity embeddings are employed. And relation embeddings are not used.

AE-TR (Averaged Entity Embeddings and TransE-based Relation Embeddings): This method represents entities as an average of their constituting word embeddings, and concatenate TransE-based relation embeddings learned from KBs to the outputs of memory networks. Both entity embeddings and relation embeddings are applied.

Table I lists the comparison results. Seen from the table, the proposed MNM outperforms all three baseline methods. Among the three baselines, the best one is AE-TR, which employs relation embeddings learned from KBs. Actually, AE-TR is similar to MNM except that entity embeddings are not learned from KBs.

Compared with AE, AE-TR simply add the relation embeddings and the *F*1-*score* is improved by 12.71%, which indicates that relation embeddings could provide effective clues about PPI relations to classifier. In MNM, both entity embeddings and relation embeddings are learned from KBs, and *F*1-*score* is 59.67%, which is 1.04% higher than AE-TR. That illustrates structured information contained in TransE-based entity embeddings is more effective than the implicit semantic information expressed by word embeddings for relation classification. However, TE makes *F*1-*score* drop 1.46% compared with AE. It perhaps because that without relation embeddings, structured prior knowledge in TransE-based entity embeddings is hard been captured by our model.

Note that if a protein entity is absent in KBs, the entity embedding is randomly initialized in MNM training. And for protein pairs not found in KBs, relation embedding is initialized as zero vectors.

TABLE I. Effects of prior knowledge

Prior knowledge	Evaluation Measurement (%)		
	p	***r***	***F*1-*score***
AE	41.33	54.47	45.92
TE	36.54	57.76	44.46
AE-TR	50.53	70.13	58.63
MNM	**50.91**	**72.50**	**59.67**

B. Sharing attention operation

Table II shows the effect of sharing attention operation. As can be seen from the table, not sharing attention operation causes *F*1-*score* to drop 0.66%. We believe that different attention operations to the two relevant entities would introduce more noises, which are not helpful to relation classification.

TABLE II. Sharing attention operation

Sharing or not	Evaluation Measurement (%)		
	p	***r***	***F*1-*score***
Not sharing	50.50	72.24	59.01
Sharing	**50.91**	**72.50**	**59.67**

C. Effects of computational layer number

We try several computational layers and the results are shown in Table III. Our MNM with different numbers of computational layers are expressed as MNM (*k*), where *k* is the number of computational layers. Among all models from single computational layer to four, we can observe that two and three layers get better performance. The best performance is achieved when the model contains two computational layers.

TABLE III. Effects of computational layer number

Computational layer number	Evaluation Measurement (%)		
	p	***r***	***F*1-*score***
MNM(*1*)	50.81	70.92	59.07
MNM(*2*)	**50.91**	72.50	**59.67**
MNM(*3*)	49.78	**74.08**	59.49
MNM(*4*)	49.61	72.63	58.83

D. Comparison with LSTM

Table IV compares our memory network-based model (MNM) with bidirectional long short-term memory (Bi-LSTM) networks [13], which is the other state-of-art networks in relation classification. Different from [13], we introduce knowledge representations learned from KBs to our Bi-LSTM model. Each entity embeddings is concatenated with context word embeddings $\{e_1, e_2, ..., e_i ..., e_n\}$ as each direction input.

Attention mechanism is applied to produce a weight vector, and merge hidden representations from each time step into a sequence-level vector. The concatenation of bidirectional sequence-level vectors is concatenated with relation

embeddings further for relation classification. From the table we can see that the proposed memory network-based model (MNM) is more powerful in taking advantage of prior knowledge than LSTM architecture.

TABLE IV. COMPARISON WITH LSTM

Model structure	Evaluation Measurement (%)		
	p	*r*	*F1-score*
Bi-LSTM	49.77	71.71	58.60
MNM	**50.91**	**72.50**	**59.67**

E. Attention visualization

To better demonstrate the effectiveness of attention mechanism, attention weights of two example sequences are visualized in the form of heat map in Fig. 2. For each sequence, the upper visible layers shows the weights of the context words toward Entity1, and the lower shows the weights of the context words toward Entity2. In the first example, "found" and "phosphorylated" have the maximum weights when paying attention to the entity "28984" and entity "207" respectively. As for the second example, "controls" has the maximum weight no matter which entity we pay attention to. From Fig. 2, we observe that our MNM could identify the important word effectively.

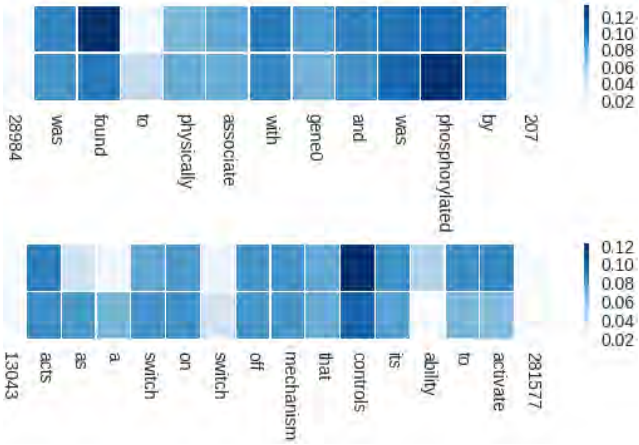


Fig. 2. Visualization of attention weight by heat map.

IV. CONCLUSION

In this paper, we propose a memory network-based PPI extraction approach. The proposed approach calculates the weight of context words toward two protein entities by leveraging prior knowledge with attention mechanism in two memory networks. The experimental results show that the proposed memory network-based approach is better than LSTM architecture, and knowledge representations learned from KBs could significantly improve the PPI performance.

ACKNOWLEDGMENT

This research is supported by National Natural Science Foundation of China (Grant No. 61772109) and The Ministry of education of Humanities and Social Science research and planning Fund of China (Grant No. 17YJA740076).

REFERENCES

1. Chowdhury, M. F., and Lavelli, A. (2012) Combining Tree Structures, Flat Features and Patterns for Biomedical Relation Extraction. conference of the european chapter of the association for computational linguistics.
2. Qian, L., and Zhou, G. (2012) Tree kernel-based protein-protein interaction extraction from biomedical literature. Journal of Biomedical Informatics, 45(3), 535-543.
3. Bui, Q., Sloot, P. M., Van Mulligen, E. M., and Kors, J. A. (2014) A novel feature-based approach to extract drug–drug interactions from biomedical text. Bioinformatics, 30(23), 3365-3371.
4. Kim, S., Liu, H., Yeganova, L., and Wilbur, W. J. (2015) Extracting drug-drug interactions from literature using a rich feature-based linear kernel approach. Journal of Biomedical Informatics, 23-30.
5. Zeng, D., Liu, K., Lai, S., Zhou, G., and Zhao, J. (2014) Relation Classification via Convolutional Deep Neural Network. international conference on computational linguistics.
6. Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998) Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.
7. Peng, Y., Lu, Z. (2017) Deep learning for extracting protein-protein interactions from biomedical literature. Proceedings of the 2017 Workshop on Biomedical Natural Language Processing.
8. Gu, J., Sun, F., Qian, L. and Zhou, G. (2017) Chemical-induced disease relation extraction via convolutional neural network. Database: The Journal of Biological Databases and Curation. 2017. 10.1093/database/bax024.
9. Schmidhuber, J. (1992) Learning complex, extended sequences using the principle of history compression. Neural Computation, 4(2):234–242.
10. Zhang, D., and Wang, D. (2015) Relation Classification via Recurrent Neural Network. arXiv: Computation and Language.
11. Zhou, H., Deng, H., Chen, L., Yang, Y., Jia, C., and Huang, D. (2016) Exploiting syntactic and semantics information for chemical–disease relation extraction. Database, 2016(2016).
12. Hochreiter, S., and Schmidhuber, J. (1997) Long Short-Term Memory. Neural Computation, 9(8), 1735-1780.
13. Zhou, P., Shi, W., Tian, J., Qi, Z., Li, B., Hao, H., and Xu, B. (2016) Attention-Based Bidirectional Long Short-Term Memory Networks for Relation Classification. meeting of the association for computational linguistics.
14. Hermjakob, H., Montecchipalazzi, L., Lewington, C., Mudali, S., Kerrien, S., Orchard, S., ... and Apweiler, R. (2004) IntAct: an open source molecular interaction database. Nucleic Acids Research, 32(90001), 452-455.
15. Stark, C., Breitkreutz, B., Reguly, T., Boucher, L., Breitkreutz, A., and Tyers, M. (2006) BioGRID: a general repository for interaction datasets. Nucleic Acids Research, 34(90001), 535-539.
16. UniProt Consortium. (2015) UniProt: a hub for protein information. Nucleic Acids Res., 43, D204–D212.
17. Bordes, A., Usunier, N., Garciaduran, A., Weston, J., and Yakhnenko, O. (2013) Translating Embeddings for Modeling Multi-relational Data. neural information processing systems.
18. Wang, Z., Zhang, J., Feng, J., and Chen, Z. (2014) Knowledge graph embedding by translating on hyperplanes. national conference on artificial intelligence.
19. Lin, Y., Liu, Z., Sun, M., Liu, Y., & Zhu, X. (2015). Learning entity and relation embeddings for knowledge graph completion. national conference on artificial intelligence.
20. Sukhbaatar, S., Szlam, A., Weston, J., and Fergus, R. D. (2015) End-to-end memory networks. neural information processing systems,, 2440-2448.
21. Tang, D., Qin, B., and Liu, T. (2016) Aspect Level Sentiment Classification with Deep Memory Network. empirical methods in natural language processing, 214-224.
22. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013) Distributed Representations of Words and Phrases and their Compositionality. neural information processing systems. 3111-3119.
23. Wei, C., Kao, H., and Lu, Z. (2015) GNormPlus: An Integrative Approach for Tagging Genes, Gene Families, and Protein Domains. BioMed Research International.
24. Kingma, D. P., and Ba, J. L. (2015) Adam: A Method for Stochastic Optimization. international conference on learning representations.

DUTIR at the BioCreative VI Precision Medicine Track: Document Triage for Identifying PPIs Affected by Genetic Mutations

Ling Luo, Zhihao Yang*, Hongfei Lin, Jian Wang
College of Computer Science and Technology, Dalian University of Technology, Dalian, China 116024

***Abstract*—The precision medicine initiative promises to identify individualized treatment depending on a patients' genetic profile and their related responses. In order to help health professionals and researchers in the precision medicine endeavor, the BioCreative VI Precision Medicine Track focuses on mining protein-protein interactions (PPIs) affected by genetic mutations from the scientific published literature. This paper describes our method used to create our submissions to the document triage subtask. In our method, several deep learning models for document triage are used, and the ensemble of them performs better than any individual model. Our best submission achieves an F-score of 0.6909 on the test set.**

Keywords—Document Triage; Precision Medicine; Deep Learning; Ensemble

I. Introduction

The precision medicine initiative (PMI) promises to identify individualized treatment depending on a patients' genetic profile and their related responses. In order to help health professionals and researchers in the precision medicine endeavor, one goal is to leverage the knowledge available in the scientific published literature and extract clinically useful information that links genes, mutations, and diseases to specialized treatments (1). Therefore, automatic text mining technique has received much attention.

Despite previous studies in protein-protein interaction (2,3) and mutation extraction (4), no one has investigated how to combine these efforts in order to help assessing and curating the clinical significance of genetic variants, an essential step towards precision medicine. Thus, the precision medicine (PM) task in BioCreative VI aims to bring together the biomedical text mining community in a new BioCreative challenge task focusing on identifying and extracting from the biomedical literature protein-protein interactions (PPIs) changed by genetic mutations. This challenge consists of two subtasks. The first subtask is Document Triage, which focuses on identifying relevant PubMed citations describing genetic mutations affecting PPIs. The second subtask is Relation Extraction. Participants in this task will be expected to build automated methods that are capable of extracting experimentally verified PPIs affected by the presence of a genetic mutation.

For the challenge, we participated in the first subtask (Document Triage Task) and our submissions to the subtask are created by the deep learning models. We presented an ensemble system that combines the results from five individual neural network models to identify relevant articles describing genetic mutations affecting PPIs from biomedical literature. The overview of our system architecture is shown in Fig. 1. Firstly, some preprocessing steps including text sentence splitting, tokenization and lowercasing are performed. Secondly, a word embedding is learned with large amounts of unlabeled data with the fastText tool (5). Moreover, the additional feature embeddings (POS and NER embedding) are introduced into the model. Then with the embeddings as input, five models are trained by the annotated training set. Finally, the results from the five models are combined by the weighted majority voting. The process is described in details in the following sections.

II. Features

Currently word embedding is widely used in the field of NLP, especially based on the deep learning methods (6). In our method, we used it as the features of our baselines. Moreover, to investigate the effects of other features (such as part of speech (POS) and named entity recognition (NER)), these features are added into the baselines as additional features. All feature embeddings are parameters of the neural network model, and they can be optimized when the model is trained. Details of each of features are presented as follows.

A. Word Embedding

Word embedding, also known as distributed word representation, can capture both the semantic and syntactic information of words from a large unlabeled corpus and has attracted considerable attention from many researchers. Compared with the bag-of-words (BOW) representation, word embedding is low-dimensional and dense. In recent years, several tools, such as word2vec (7) and fastText (5), have been widely used in the field of NLP. To achieve a high-quality word embedding, we downloaded a total of 1,322,107 MEDLINE abstracts from the PubMed website with the query string "protein" as the unlabeled data. Then the data and the training dataset (a total of 4,082 abstracts) provided in the PM document triage task were used to train 50-dimensional word embedding by the fastText tool as pre-trained word embedding.

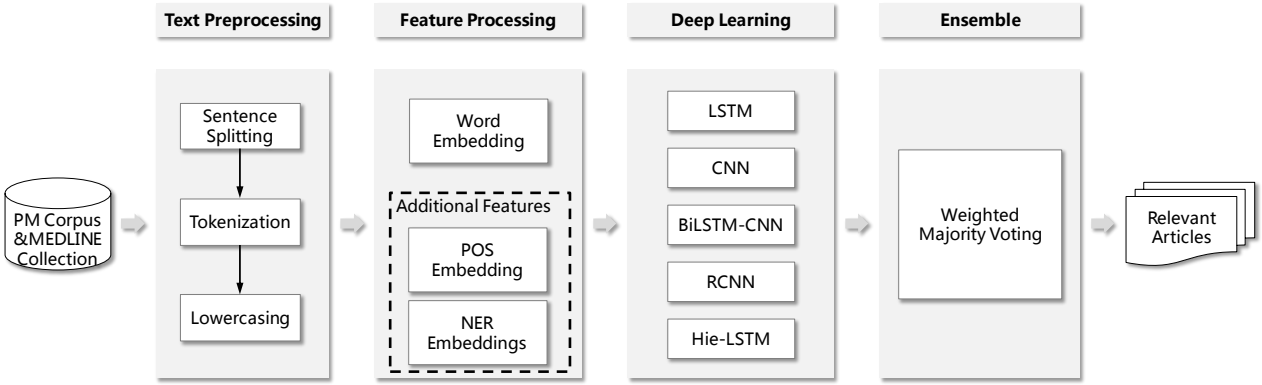


Fig. 1. The overview of the system architecture.

B. Additional Features

Due to the complexity of the natural language, some linguistic features are often employed in traditional machine learning methods. We also explored the effect of additional features (such as POS and NER). The POS information of each word were generated by the Stanford CoreNLP tool (8). In addition, NER tags information (including gene, chemical, disease and mutation entities) generated by the PubTator tagger (9) was also used as a feature. The dimensions of the POS and NER embedding are both five and they were initialized randomly.

III. Description of the Models

In this section we describe in details the five models used in our system, which include 1) LSTM, 2) CNN, 3) BiLSTM-CNN, 4) RCNN and 5) HieLSTM.

A. LSTM

A recurrent neural network model, namely long-short term memory (LSTM) (10), is used in our ensemble system. For the tth word in the article, an LSTM takes as input $\mathbf{x}_t$, and produces $\mathbf{h}_t$ based on the following formulas:

$$\mathbf{i_t} = \sigma(\mathbf{W}_{xi}\mathbf{x}_t + \mathbf{W}_{hi}\mathbf{h}_{t\text{-}1} + \mathbf{W}_{ci}\mathbf{c}_{t\text{-}1} + \mathbf{b}_i) \quad (1)$$

$$\mathbf{c}_t = (1-\mathbf{i}_t) \square \ \mathbf{c}_{t-1} + \mathbf{i}_t \square \ \tanh(\mathbf{W}_{xc}\mathbf{x}_t + \mathbf{W}_{hc}\mathbf{h}_{t-1} + \mathbf{b}_c) \quad (2)$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_{xo}\mathbf{x}_t + \mathbf{W}_{ho}\mathbf{h}_{t-1} + \mathbf{W}_{co}\mathbf{c}_t + \mathbf{b}_o) \quad (3)$$

$$\mathbf{h}_t = \mathbf{o}_t \square \ \tanh(\mathbf{c}_t) \quad (4)$$

where σ is the element-wise sigmoid function, and $\square$ is the element-wise product. $\{\mathbf{W}_{ij}\}$ is the weight matrix set. $\{\mathbf{b}_i\}$ is the bias vector set.

Then the sequence of vectors $\mathbf{h}_{1:n}$ output from all LSTM cells are combined into a single vector that represents the article by the max pooling layer. At last, a softmax function is used on this feature vector to compute the predictive probabilities of the article types.

B. CNN

In the convolutional neural network (CNN) model, a convolution operation is applied to produce local features. Given an input sequence $\mathbf{X}=[\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_i,\ldots,\mathbf{x}_n]$, a fixed size k window approach is used to capture each element's context information. Then a matrix operation, as shown in formula (5), is applied to each successive window in the sequence:

$$\mathbf{C} = \mathrm{ReLU}(\mathbf{W}_{con} \bullet \mathbf{X}_{t:t+k-1} + \mathbf{b}_f) \quad (5)$$

where $\mathbf{W}_{con}$ is the transformation matrix that is the same across all windows in the article, ReLU is the rectified linear unit function (11), and $\mathbf{C}$ is the convolutional layer result.

In our CNN model, two consecutive convolutional layers (window size k=3) are stacked to extract convolutional features. Then a max pooling operation and a fully connected hidden layer are used after convolutional features. Finally, a softmax function is used to classify articles.

C. BiLSTM-CNN

In general, CNN is capable of extracting local information and LSTM can capture long-distance dependency. So we combined the two neural network architectures into a model, namely BiLSTM-CNN. The model consist two parts, a bidirectional LSTM (BiLSTM) layer and a convolution layer. Firstly, an article is represented as a sequence of embeddings by the embedding layer. Next, the embeddings are given as input to a BiLSTM layer. In the BiLSTM layer, a forward LSTM computes a representation of the sequence from left to right, and another backward LSTM computes a representation of the same sequence in reverse. These two distinct networks use different parameters, and then the representation of a word is obtained by concatenating its left and right context representations. Then a tanh layer on top of the BiLSTM is used to learn higher features. Next the features are fed into a convolution layer and a max pooling operation is used to extract global features from the convolution layer. Finally, a softmax function is used to classify articles.

D. RCNN

Similar with the BiLSTM-CNN model, Lai et al. proposed a recurrent convolutional neural network (RCNN) for

TABLE I. PERFORMANCE OF INDIVIDUAL MODELS ON OUR DEVELOPMENT SET

Model	*Precision*	*Recall*	*F-score*	*ACC*
LSTM	0.7157	0.8650	**0.7833**	**0.8088**
CNN	0.6784	0.8282	0.7459	0.7745
BiLSTM-CNN	0.6908	0.8773	0.7730	0.7941
RCNN	0.6878	0.8650	0.7663	0.7892
Hie-LSTM	0.7181	0.8282	0.7692	0.8015

document classification (12). In the model, the recurrent structure to capture the contextual information to the greatest extent possible when learning word representations of articles, which may introduce considerably less noise compared to a traditional window-based neural network. Moreover, the model can reserve a large range of the word ordering when learning representations of articles. A max pooling layer is also employed that automatically judges which features in articles play key roles.

E. Hie-LSTM

Recently, Yang et al. proposed a hierarchical attention network (HAN) for document classification (13). The model is designed to capture two basic insights about document structure. First, since documents have a hierarchical structure (words form sentences, sentences form a document), a document representation is constructed by first building representations of sentences and then those are aggregated into a document representation. Second, it is observed that different words and sentences in a document are differentially informative.

Our implementation of the model (Hie-LSTM) is similar to HAN, which has a hierarchical structure of documents using two LSTMs instead of two bidirectional GRUs in HAN. And the max-pooling layer is used in our model instead of the original attention layer in HAN.

IV. MODEL ENSEMBLE

In our method, the results of the above mentioned models are combined via the weighted majority voting (14). In the method, only the class labels are available from the classifier outputs. We define the decision of the tth classifier as $d_{t,j} \in \{0,1\}$. The ensemble result then chooses class J that receives the largest total vote:

$$\sum_{t=1}^{T} w_t d_{t,J} = \max_{j=1}^{C} \sum_{t=1}^{T} w_t d_{t,j} \tag{6}$$

where w_t is the weight of classifier t, $\sum w_t=1$ and $d_{t,j}$ is 1 or 0 depending on whether classifier t chooses j or not.

We found the best setting of weights via brute force grid search, quantizing the coefficient values in the interval [0, 1] at increments of 0.1. The search was evaluated on our development set to avoid overfitting.

TABLE II. THE EFFECT OF ADDITIONAL FEATURES ON PERFORMANCE ON OUR DEVELOPMENT SET

Model	*Precision*	*Recall*	*F-score*	*ACC*
LSTM	0.7157	0.8650	**0.7833**	**0.8088**
+POS	0.6441	0.8773	0.7429	0.7574
+NER	0.6460	0.8957	0.7506	0.7623
+POS+NER	0.6307	0.9325	0.7525	0.7549
CNN	0.6784	0.8282	**0.7459**	**0.7745**
+POS	0.6650	0.8160	0.7328	0.7623
+NER	0.6250	0.9203	0.7444	0.7476
+POS+NER	0.6717	0.8160	0.7368	0.7672
BiLSTM-CNN	0.6908	0.8773	**0.7730**	**0.7941**
+POS	0.7167	0.7914	0.7522	0.7917
+NER	0.6904	0.8344	0.7556	0.7843
+POS+NER	0.6636	0.8712	0.7533	0.7721

V. RESULTS

The organizers of the BioCreative VI PM document triage tasks provided a corpus including the training and test sets. The training set consists of 4,082 annotated PubMed articles (title and abstract) as relevant or not relevant, and the test set consists of 1,500 unannotated articles. In our experiments, we randomly selected the 10% of the training set as the development set to tune the hyper-parameters. And the document triage performance was measured with an F-score which attributes equal importance to precision and recall (F1 score). In addition, an Accuracy (Acc) measure is also used to evaluate the performance.

A. Training settings

In our method, the parameters of the model in the word embedding are initialized with pre-trained word embeddings and other parameters are initialized at random from a uniform distribution. Then all parameters of models (except for the CNN using Adadelta (15)) are optimized using RMSprop (16) to minimize categorical cross-entropy. Our models were implemented using open-source deep learning library keras (https://keras.io) and trained on a NVIDIA Tesla K40 GPU.

B. Performance of Individual Models

Table I reports the results of each individual model on our development set. We found that the CNN model performed the worst, with BiLSTM-CNN slightly better than RCNN and Hie-LSTM. The most competitive model is the LSTM model.

C. The Effect of Additional Features on Performance

We also investigated the effect of two additional features (POS and NER embeddings mentioned in the section II.*B*) on the performance of the models (LSTM, CNN and BiLSTM-CNN) and Table II shows the results of different combinations of these features.

The results show that F-scores generally decrease when these additional features are added into the models. When only POS feature is added, the models perform the worst. One

TABLE III. PERFORMANCE OF DIFFERENT MODEL COMBINATIONS ON OUR DEVELOPMENT SET AND THE TEST SET

Model		*Our Development Set*				*Official Test Set*			
	Weighted	*Precision*	*Recall*	*F-score*	*ACC*	*Precision*	*Recall*	*F-score*	*ACC*
LSTM	1.0	0.7157	0.8650	0.7833	0.8088	0.5890	0.8068	0.6809	0.6687
LSTM	0.2								
CNN	0.1								
BiLSTM-CNN	0.3	0.7398	0.8896	0.8078	0.8309	0.6112	0.7945	**0.6909**	**0.7284**
RCNN	0.1								
Hie-LSTM	0.3								
LSTM	0.3								
CNN$^+$	0.1								
BiLSTM-CNN	0.3	0.7436	0.8896	**0.8101**	**0.8333**	0.5882	0.8219	0.6857	0.7103
BiLSTM-CNN$^+$	0.1								
Hie-LSTM	0.2								

$^+$means the model using the additional POS and NER embedding

plausible reason is that POS information is not necessary for the biomedical document triage task. Moreover, the noise may be introduced into the model by the errors of the POS and NER tools.

D. Performance of Different Model Combinations

Finally, the results of the previous models are combined into an ensemble. The results of our submitted runs on our development set and official results of the runs on the test set are shown in Table III. Our best submission achieves an F-score of 0.6909 on the test set. On our development set, although F-scores generally decrease when the additional features are added into the models, they can help boost the performance of the ensemble system. When the results of LSTM, BiLSTM-CNN, Hie-LSTM, CNN with additional features and BiLSTM-CNN with additional features are combined, the highest F-score of 0.8101 is achieved on our development set. Moreover, the weights of models show that LSTM, BiLSTM-CNN and Hie-LSTM models have more contribution for the overall ensemble performance.

ACKNOWLEDGMENT

This work is supported by the grants from the Natural Science Foundation of China (No. 61272373, 61572102 and 61572098), Trans-Century Training Program Foundation for the Talents by the Ministry of Education of China (NCET-13-0084), and the Major State Research Development Program of China (No. 2016YFC0901902).

REFERENCES

1. Singhal, A., Simmons, M., Lu, Z. (2016) Text mining genotype-phenotype relationships from biomedical literature for database curation and precision medicine. PLoS computational biology, 12, e1005017.

2. Kim, S., Islamaj Doğan, R., Chatr-Aryamontri, A., et al. (2016) BioCreative V BioC track overview: collaborative biocurator assistant task for BioGRID. Database, 2016, baw121.

3. Krallinger, M., Vazquez, M., Leitner, F., et al. (2011) The Protein-Protein Interaction tasks of BioCreative III: classification/ranking of articles and linking bio-ontology concepts to full text. BMC bioinformatics, 12, S3.

4. Wei, C.-H., Harris, B.R., Kao, H.-Y., Lu, Z. (2013) tmVar: a text mining approach for extracting sequence variants in biomedical literature. Bioinformatics, 29, 1433-1439.

5. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T. (2016) Enriching word vectors with subword information. arXiv preprint arXiv:1607.04606.

6. Lai, S., Liu, K., Xu, L., Zhao, J. (2015) How to Generate a Good Word Embedding? arXiv preprint arXiv:1507.05523.

7. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J. (2013) Distributed representations of words and phrases and their compositionality. Advances in neural information processing systems, pp. 3111-3119.

8. Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J.R., Bethard, S., McClosky, D. (2014) The stanford corenlp natural language processing toolkit. ACL (System Demonstrations), pp. 55-60.

9. Wei, C.-H., Kao, H.-Y., Lu, Z. (2013) PubTator: a web-based text mining tool for assisting biocuration. Nucleic acids research, 41, W518-W522.

10. Hochreiter, S., Schmidhuber, J. (1997) Long short-term memory. Neural computation, 9, 1735-1780.

11. Glorot, X., Bordes, A., Bengio, Y. (2011) Deep Sparse Rectifier Neural Networks. Aistats, vol. 15, pp. 275.

12. Lai, S., Xu, L., Liu, K., Zhao, J. (2015) Recurrent Convolutional Neural Networks for Text Classification. AAAI, vol. 333, pp. 2267-2273.

13. Yang, Z., Yang, D., Dyer, C., He, X., Smola, A.J., Hovy, E.H. (2016) Hierarchical Attention Networks for Document Classification. HLT-NAACL, pp. 1480-1489.

14. Polikar, R. (2006) Ensemble based systems in decision making. IEEE Circuits and systems magazine, 6, 21-45.

15. Zeiler, M.D. (2012) ADADELTA: an adaptive learning rate method. arXiv preprint arXiv:1212.5701.

16. Hinton, G., Srivastava, N., Swersky, K. (2012) RMSProp: Divide the gradient by a running average of its recent magnitude. Neural networks for machine learning, Coursera lecture 6e.

Identifying Mutation-induced Protein-Protein Interactions in Scientific Literature

Eric Zhe-You Chen[1], Onkar Singh[2,3], Toni Rose Jue[4], Chen-Kai Wang[5], Jitendra Jonnagaddala[4,6], Hong-Jie Dai[1*]

[1]Department of Computer Science and Information Engineering, National Taitung University, Taitung, Taiwan
[2]Bioinformatics Program, Taiwan International Graduate Program, Institute of Information Science, Academia Sinica, Taipei, Taiwan
[3]Institute of Biomedical Informatics, National Yang-Ming University, Taipei, Taiwan
[4]Prince of Wales Clinical School, UNSW Sydney, Australia
[5]Graduate Institute of Biomedical Informatics, Taipei Medical University, Taipei, Taiwan
[6]School of Public Health and Community Medicine, UNSW Sydney, Australia

***Abstract*—**Identifying mutation induced protein-protein interaction (PPI) is a prime concern in the advancement of precision medicine (PM). The aim of this study is to develop an automatic tool to identify mutation-induced PPI from biomedical literature. Identification of such interactions is important for understanding the complex network of pathways that make up heterogeneous diseases. In the work, we applied two machine learning algorithms to deal with the problem. The first is the support vector machine. We proposed features including *n*-gram and article-meta information and train two SVM models with the linear kernel. The second is the neural network. We proposed a new network structure based on the convolutional neural network (CNN) which integrates convolved context features from different paragraphs and handcrafted features for MeSH term information. The performance of the developed models was evaluated on the training set of the BioCreative VI PM document triage task. The SVM-based approaches achieved the best overall F-score, while CNN models have better precision.In the future, we will release the tool and continue to improve the performance of the developed CNN models by replacing the randomized initialized embedding layer with pre-trained word vectors and including more meta-information with.

Keywords— precision medicine, personalized medicine, targeted therapeutics, CNN, PPI

I. Introduction

Precision medicine (PM) has been founded on the concept of specifically designing treatment based on molecular targets. Owing to the recent advances in high-throughput screening of patients' tumors, disease prevention and treatment can be tailored based on the individual patient's genetic profile (Pavelić, Martinović, & Kraljević Pavelić, 2015; Tannock & Hickman, 2016). The "one size, fits all" approach is no longer feasible, especially for patients diagnosed with heterogeneous diseases.

However, a gap still exists in the process of identifying these molecular targets and how they function within pathways. The unprecedented growth of biomedical literature provides a unique opportunity to bridge this existing gap. Biomedical literature can provide valuable information of the functional influence of genetic variations and alterations to protein-protein interactions (PPIs). Identification of such interactions is important for understanding the complex network of pathways that make up heterogeneous diseases. An example is in glioblastoma (GBM). GBM is a devastating malignant tumor in the brain with a highly heterogeneous characteristic found between and within patient tumors (T. Jue, Hovey, Davis, Carleton, & McDonald, 2015; T. R. Jue & McDonald, 2016). Due to high heterogeneity, treatment options have been limited for GBM patients. Manually identifying molecular/genetic targets and their interaction in pathways from publicly available literature has been cost-, labor- and time-intensive. Text mining becomes an inevitable technique in identifying significant genotype-phenotype data buried in the biomedical literature that can be translated into clinical use. Text mining can impact the drug discovery process, especially in heterogeneous diseases such as GBM.

In this study, we present our text mining efforts to automatically identify PubMed articles with genetic mutations affecting PPIs using the dataset published by (Doğan et al., 2017). We propose several features and train them with the support vector machine (SVM) (Vapnik, 1995). On the other hand, considering the recent resurgence in neural network, which provides a way to avoid the complicated process of feature engineering in supervised natural language processing tasks such as text classification(Rios & Kavuluru, 2017), named entity recognition (Lample, Ballesteros, Subramanian, Kawakami, & Dyer, 2016) and machine translation (Sutskever, Vinyals, & Le, 2014), we apply the convolutional neural network (CNN) architecture and study its performance for the task of identifying relevant PubMed articles that demonstrate the role of genetic mutations in changing PPIs. CNN model has achieved great success in the image classification challenge (Krizhevsky, Sutskever, & Hinton, 2012) and widely applied in several text classification tasks (Hsieh et al., 2016; Kim, 2014;

Zhang & LeCun, 2015; Zhang, Zhao, & LeCun, 2015). Based on the fundamental network architecture, we propose a new neural network architecture for our application and compare its performance with the SVM models and traditional CNN architecture.

II. Methods

A. SVM-based Systems

We first present the systems developed base on SVM. First of all, for each article, we used the NCBI E-Utilities to determine the MeSH terms associated with it. In total of 5,376 unique MeSH terms were collected. The author names and their affiliation information were also collected as the meta-information of the given article. We then tokenized the title and the abstract of each article by using GENIA tagger (Tsuruoka et al., 2005) and removed tokens which were listed as stop words in the PubMed stop word[1].

For each article, we extract the following features and used the sequential minimal optimization algorithm to train our SVM models with the linear kernel.

- *n*-gram (*n*=1~3) features: We used TF-IDF as the weighing scheme to extract the features
- MeSH term information: The MeSH terms assigned to the article are extracted as features.
- Author names and their affiliation information: We derive two novel feature sets. First, the author names and their affiliation information are listed as two nominal features. For the second feature set, we use cosine similarity to calculate a similarity score between authors' department details, title and abstract, and represent the scores as a numerical feature.

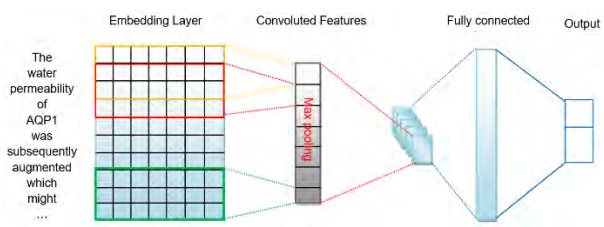


Fig. 1. The fundamental CNN network architecture for bindary calssfication task.

B. Fundamental CNN Architecture for Text Classification

CNN is a feed-forward neural network with convolution layers interleaved with pooling layers. Fig. 1 shows the fundamental network architecture employed by most of the CNN-based text classification works. The input layer is composed of the word sequence appearing in the given text. For our task, the existing words in a given PubMed abstract form the input layer. The word sequence can be represented as a word matrix, in which each row is a vector corresponding to one word appearing in the sequence in one-hot form, and the dimension of the vector is equal to the vocabulary size determined by the number of unique words of the given dataset.

[1]https://www.ncbi.nlm.nih.gov/books/NBK3827/table/pubmedhelp.T.stopwords/

These vectors are mapped to low-dimensional representations through a matrix product which is illustrated as the word embedding layer in Fig1.

The embedding layer then goes through the convolution layer, which applies convolutions over the embedded word input to compute its output. Let $x_i \in \mathrm{R}^k$ be the k-dimensional word embedding vector corresponding to the *i*th word of the input sequence. The input text **x** of length *n* can be represented as

$$x_{1:n} = x_1 \oplus x_2 \oplus \ldots \oplus x_n \tag{1}$$

A convolution can be consider as executing a sliding window function *f* that slides over full rows of the given word matrix. We use *h* to indicate the size of the window. Therefore, the word vectors of a context window with a length of *h* can be represented as $x_{i:i+h-1}$, where $x_{i:i+h-1}$ refer to the concatenation of words $x_i, x_{i+1}, \ldots, x_{i+h-1}$. This sliding window function *f* as defined in (2) serves the similar function of the kernel during image processing for extracting features.

$$c_i = f(\mathbf{w} \cdot x_{i:i+h-1} + b) \tag{2}$$

Equation (2) shows the feature c_i generated from a window of words $x_{i:i+h-1}$. Here *b* is a bias value and *f* is a non-linear function. The function is known as a filter or convolution matrix in image processing.

This function is applied to each possible window in the input sequence $\{x_{1:h}, x_{2:h+1}, \ldots, x_{n-h+1:n}\}$ by multiplying its values element-wise with the words in the window and sums them up. It then slides over the whole matrix to obtain the convolution result, which is referred to as a feature map as defined in (3).

$$\mathbf{c} = [c_1, c_2, \ldots, c_{n-h+1}] \tag{3}$$

The convolution step indicated in Fig. 1 connects each region composing of three-word vectors of the input matrix to oneoutput neuron.For each feature map, we apply the max-over-time pooling operation to output the maximum activation value (4).

$$\hat{c} = \max(\mathbf{c}) \tag{4}$$

Therefore, the max-pooling layer following the convolutional layer can subsample the feature maps and induce a fixed-length vector composed of the most important features.

Finally, similar to regular neural networks, all outputs from the max-pooling layer are concatenated to form a fixed-length feature vector, which is then fully connected to a softmax layer for generating an output with a probability distribution over the possible classes.

C. Proposed CNN Architecture

This section delineates the proposed CNN architecture for the mutation-induced PPI document triage task. Considering the input of our classification system is a biomedical abstract, we enhance the network architecture presented in Fig. 1 from

two ways. First we decorate the max-pooled vectors with article meta-information. In this study, we leveraged the MeSH term information associated with each article to enhance the max pooled vector. We represent the feature vector of the MeSH terms by the one-hot encoding scheme. Because the numbers of unique MeSH terms observed in the training set is 5,376, the size of the feature vector encoding all MeSH terms is 5,376.

Furthermore, sentences described in abstracts of biomedical papers could be sequentially divided into sections like objective, methods, results and conclusion(Lin, Dai, Bow, Chiu, & Tsai, 2009) and the words used in each of which may convey different semantics. We therefore propose to capture the information in different paragraphs by using different CNNs. The max-pooled vectors generated by all of the CNNs are then concatenated with each other to form a dense vector. The vector is finally fully connected to a softmax layer for the classification outcome. Fig. 2 shows the proposed CNN architecture developed for our application.

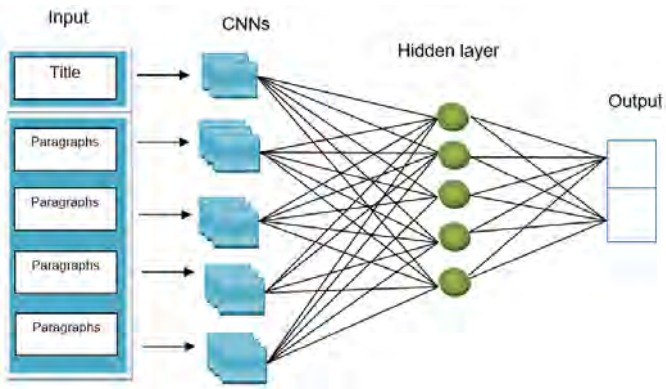


Fig. 2. The proposed CNN network architecture for the document triage task.

As shown in Fig. 2, we construct five CNNs in our network. We connect the nodes representing the word sequence from the article title to the first CNN only. For every two sentences sequentially appeared in the abstract, we connect them to the second to fourth CNNs. For the remaining sentences, we connect them to the last CNN.

III. Results

A. Experimental Setup

The training set from (Doğan et al., 2017)was used in this study. It contains 1,729 positive instances and 2,253 negative instances.Various configurations' performance was assessed in this study. "Basic CNN" configuration refers to the standard CNN architecture illustrated in Fig. 1. The "proposed CNN" configuration is the network architecture contains the five CNNs as depicted in Fig. 2. The "Proposed CNN + MeSH" is the multi-CNN architecture along with the MeSH information.

In the "SVM Baseline" configuration, the training set is generated by removing stop-words. This configuration was extended by further including the MeSH term information associated with each article (indicated as "SVM Baseline + MeSH"). Finally, the "SVM Baseline + MeSH + Similarity" configuration includes all features described earlier. The second SVM-based system, denoted as "SVM Baseline (NER)", was developed by removing sentences without containing any entities mentions such as genes, mutations, and chemical names. We used the recognition results of PubTator (Wei, Kao, & Lu, 2013) to conduct the filtering process. Furthermore, all entity mentions recognized by PubTator were replaced with their categorization tags in the training set.

We applied three-fold cross validation on the training set. For all experiments related to CNN, we used a word vector of size 300. The value of each word vector is randomly initialized. We set the context window and number of feature maps as 3 and 50, respectively. The toolkit used for implementing the CNN model is CNTK[5]. We used Weka to implement the model trained with SVM algorithms. The standard metrics including precision (P), recall (R) and F-measure (F) were used to report the performance of the developed models.

B. Experimental Results

Table 1 presents the results on the training set. The best precision among all studied methods is 0.627 which can be attributed to the basic CNN model. Comparing with the basic network architecture, the proposed CNN architecture has better recall and results in an improved F-score by 0.008. By decorating the proposed architecture with the MeSH term information, the precision can be further improved to 0.613 leading to a better F-score. We can learn from the results that the proposed network architecture has a better generalization power while the basic CNN model can precisely identify documents related to mutations affecting PPIs.

TABLE I. Three Fold Cross Validation Restuls on The Training Set

Configuration	P	R	F
Basic CNN	**0.627**	0.618	0.623
Proposed CNN	0.604	0.659	0.631
Proposed CNN+MeSH	0.613	0.655	0.633
SVM Baseline	0.592	**0.800**	0.680
SVM Baseline + MeSH	0.607	0.790	0.686
SVM Baseline + MeSH + Similarity	0.615	0.778	**0.687**
SVM Baseline (NER)	0.603	0.751	0.669
SVM Baseline (NER) + MeSH	0.614	0.751	0.676

The performance of SVM-based approaches has better overall F-scores but lower precisions. The best recall (0.8) was achieved by the SVM baseline. We can also recognize that the precision of the SVM-based methods can be improved with the inclusion of more handcraft features. The best precision achieved by SVM is 0.615, which is attributed to the model with all proposed features. The results demonstrated the capability of the SVM models for generalizing the context relevant to mutations affecting PPIs is better that the studied CNN models. However, the models require more handcraft features to achieve a better precision.

Finally, we can see that the SVM-based models trained with NER information have lower F-scores. After analyzing the results, we believe this is due to that some abstracts of the

[5]https://www.cntk.ai/

training set haven't be processed by PubTator, therefore no entity information is available for them.

IV. Conclusion

In this study, we applied two popular text classification algorithms for automatically identify PubMed articles with genetic mutations affecting PPIs. In addition to the basic *n*-gram features, we proposed to encode article-meta information including MeSH terms and authorship as features to improve the performance of SVM. The three-fold cross validation on the training set demonstrated that the proposed features can improve both the precision and F-scores. We also adapted the traditional CNN architecture for our task. Comparing with SVM, the CNN-based model had a better precision but lower precision and F-measure. We believed that it may be due to the parameter sharing scheme used in the traditional CNN model. We therefore proposed new network architecture to capture different semantic features buried in different paragraphs. The proposed architecture can also encode handcrafted features like MeSH term information. Compared the performance of the proposed architecture with that of the traditional CNN, the results demonstrated the proposed architecture can improve the recall and F-measure of the traditional CNN architecture. In the future, we will release the tool and continue to improve the performance of the developed CNN models by replacing the randomized initialized embedding layer with pre-trained word vectors. We will also include more meta-information and incorporate techniques for embedding author and MeSH term information into low-dimensional vector spaces to provide better representations.

References

Doğan, R. I., Chatr-aryamontri, A., Kim, S., Wei, C.-H., Peng, Y., Comeau, D. C., & Lu, Z. (2017). *BioCreative VI Precision Medicine Track: creating a training corpus for mining protein-protein interactions affected by mutations*. Paper presented at the Proceedings of the BioNLP 2017 workshop, Vancouver, Canada.

Hsieh, Y.-L., Chang, N.-W., Chen, K.-Y., Huang, Y.-C., Tsai, W.-Y., Jonnagaddala, J., . . . Hsu, W.-L. (2016). *Deep Convolutional Neural Network Document Model for Classifying Symptom Severity in an Research Domain Criteria domain*. Paper presented at the Proceedings of the 2016 CEGS NGRID SharedTasks and Workshop on Challenges in Natural Language Processing for Clinical Data.

Jue, T., Hovey, E., Davis, S., Carleton, O., & McDonald, K. (2015). Incorporation of biomarkers in phase II studies of recurrent glioblastoma. *Tumor Biology, 36*(1), 153-162. doi:10.1007/s13277-014-2960-3

Jue, T. R., & McDonald, K. L. (2016). The challenges associated with molecular targeted therapies for glioblastoma. *Journal of Neuro-Oncology, 127*(3), 427-434.

Kim, Y. (2014). *Convolutional neural networks for sentence classification*. Paper presented at the Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). *Imagenet classification with deep convolutional neural networks.* Paper presented at the ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS.

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., & Dyer, C. (2016). *Neural architectures for named entity recognition*. Paper presented at the Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies

Lin, R. T. K., Dai, H.-J., Bow, Y.-Y., Chiu, J. L.-T., & Tsai, R. T.-H. (2009). Using conditional random fields for result identification in biomedical abstracts *Integrated Computer-Aided Engineering, 16*(4), 339-352.

Pavelić, K., Martinović, T., & Kraljević Pavelić, S. (2015). Do we understand the personalized medicine paradigm?: Personalized medicine marks the beginning of a new attitude in medicine. *EMBO Reports, 16*(2), 133-136.

Rios, A., & Kavuluru, R. (2017). Ordinal convolutional neural networks for predicting RDoC positive valence psychiatric symptom severity scores. *J Biomed Inform*. doi:10.1016/j.jbi.2017.05.008

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). *Sequence to sequence learning with neural networks.* Paper presented at the ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS.

Tannock, I. F., & Hickman, J. A. (2016). Limits to Personalized Cancer Medicine. *New England Journal of Medicine, 375*(13), 1289-1294.

Tsuruoka, Y., Tateishi, Y., Kim, J.-D., Ohta, T., McNaught, J., Ananiadou, S., & Tsujii, J. i. (2005). Developing a robust part-of-speech tagger for biomedical text. *Lecture notes in computer science*, 382-392.

Vapnik, V. N. (1995). *The nature of statistical learning theory*. Berlin: Springer-Verlag New York, Inc.

Wei, C.-H., Kao, H.-Y., & Lu, Z. (2013). PubTator: a web-based text mining tool for assisting biocuration. *Nucleic Acids Research, 41*(W1), W518-W522. doi:10.1093/nar/gkt441

Zhang, X., & LeCun, Y. (2015). Text understanding from scratch. *arXiv preprint arXiv:1502.01710*.

Zhang, X., Zhao, J., & LeCun, Y. (2015). *Character-level convolutional networks for text classification.* Paper presented at the ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS.

MayoNLP at the BioCreative VI PM Track: Entity-enhanced Hierarchical Attention Neural Networks for Mining Protein Interactions from Biomedical Text

Yanshan Wang, Feichen Shen, Ravikumar Komandur Elayavilli, Sijia Liu, Majid Rastegar-Mojarad, Hongfang Liu
Department of Health Sciences Research, Mayo Clinic, Rochester, MN, USA

***Abstract*—This paper describes our participation in both tasks of the BioCreative VI Precision Medicine track: Document Triage Task and Relation Extraction Task. We propose Entity-enhanced Hierarchical Attention Neural Networks (EHANN) for tackling both tasks. The EHANN model leverages three attention mechanisms to combine and differentiate three different kinds of information from the biomedical texts, i.e., word, entity, and sentence, when constructing a document representation. The proposed model was evaluated on the provided training data using 10-fold cross-validation. We achieved average accuracy of 96.65% for the Document Triage Task and that of 96.31% for the Relation Extraction Task. The experimental results show that the EHANN is an effective model for mining protein interactions from biomedical texts.**

Keywords—neural networks; deep learning; attention mechanism; named entity recognition; protein-protein interactions

I. Introduction

In 2015 President Obama announced a research initiative that aims to accelerate progress toward a new era of precision medicine (PM). With the recent development of large-scale biologic databases, powerful methods for characterizing patients, and computational tools for analyzing large sets of data, the prospect of applying the concept of PM broadly has been dramatically improved (1). Since a large amount of knowledge is available in the scientific published literature, extracting clinically useful information, such as relations between genes, mutations, and diseases, from the literature could help health providers make specialized decisions for diagnosis and treatments (2). Discovering the knowledge of protein-protein interaction (PPI) changed by genetic mutations from the biomedical literature is an essential step towards PM since how genetic background influence signaling pathways of PPI is crucial for predicting disease phenotypes (3).

BioCreative VI PM track is a new BioCreative challenge task focusing on identifying and extracting from the biomedical literature protein-protein interactions changed by genetic mutations. This track has two tasks: 1) identifying relevant PubMed articles describing genetic mutations affecting PPIs (*Document Triage Task*), and 2) extracting experimentally verified PPI affected by the presence of a genetic mutation from PubMed articles (*Relation Extraction Task*). In the *Document Triage Task*, the organizer provide a training dataset that contains PubMed articles (title and abstract) with manually created labels (relevant/nonrelevant) by BioGRID database curators and participants need to implement automatic methods capable of return a relevance-ranked judgement for a list of testing Pubmed articles. In the *Relation Extraction Task*, a training dataset is also provided with manually annotated relevant interacting protein pairs. Given a testing data set (PubMed title and abstract), participants need to firstly recognize and identify gene/proteins mentioned in the text and normalize them to Entrez Gene ID, and secondly return the set of interacting protein pairs (and their corresponding Gene Entrez IDs) mentioned in the text that are affected by a genetic mutation.

This paper describes our participation in both tasks. We view the first task as a classification task and the second as a combination of a named entity recognition task and a classification task. We propose Entity-enhanced Hierarchical Attention Neural Networks (EHANN), a novel neural architecture, for the classification tasks. EHANN is an extension to the Hierarchical Attention Network (HAN) (4). HAN includes two attention mechanisms at the word and sentence level so that the model could pay more or less attention to individual words and sentences when constructing the representation of a document. Different from the HAN, the proposed EHANN constructs a document representation by aggregating representations of entities in addition to word and sentence representations as in HAN. This will enable EHANN to especially capture entity relation information beyond word and sentence information. Moreover, the proposed EHANN leverages one more attention mechanism at the entity level in addition to the two attention mechanisms so that different entities are differentially treated. We leveraged BELminer, a previously developed system (5), for extracting entities in both tasks.

II. Methods

A. Entity-enhanced Hierarchical Attention Neural Networks

The architecture of the proposed EHANN is shown in Figure 1. It consists of eight neural network layers (from bottom to top, they are word/entity representation layer, bidirectional gated recurrent unit (GRU) layer, attention layer, sentence and entity representation layer, bidirectional GRU layer, attention layer, document representation layer, and classification layer), which can be categorized into five main parts, namely, word/entity sequence encoder, word/entity level attention layer, sentence/entity-bunch encoder, sentence and entity-bunch level attention layer, classification layer. At the beginning, the word or entity sequences are fed into a word or

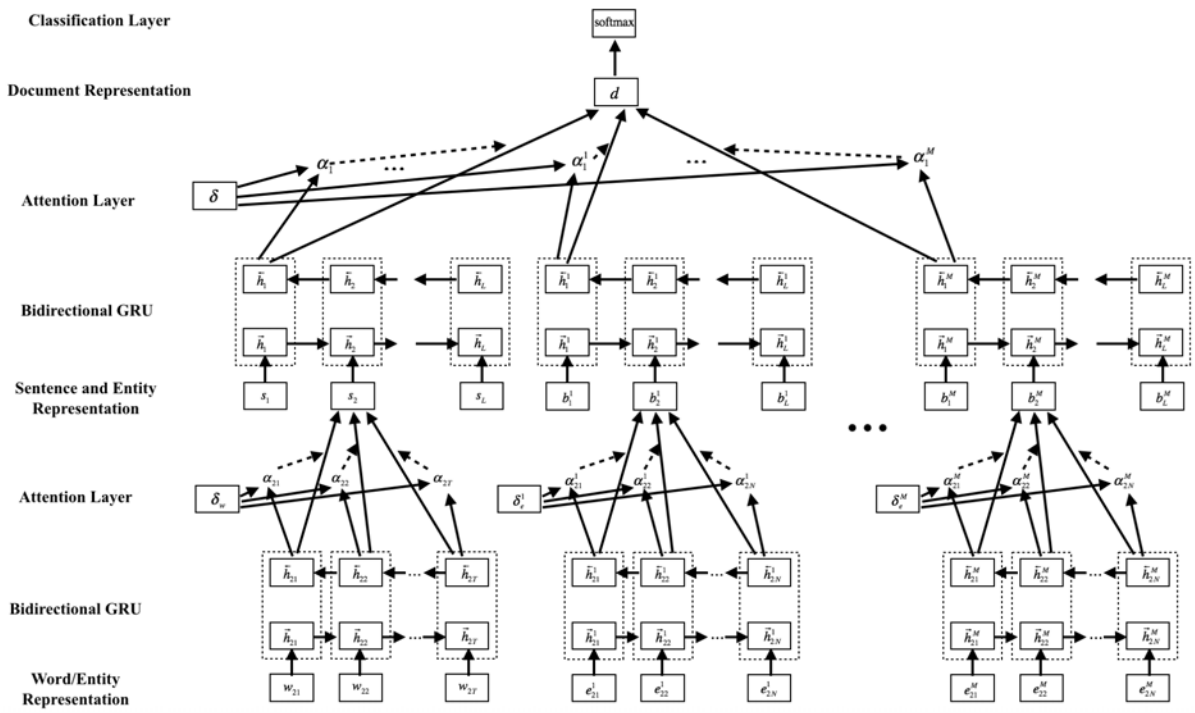


Fig. 1. Architecture of Entity-enhanced Hierarchical Attention Neural Networks.

entity sequence encoder and represented by a word or entity annotation. Subsequently, word or entity level attention mechanisms take a word or entity annotation as input and output a sentence or entity-bunch (we call a combination of entities from a sentence as an *entity-bunch*) representation. Then the sentence or entity-bunch representation is fed into another attention mechanism simultaneously and generates a document representation. Eventually, a softmax function layer is used on the document representation for classification. We explain each part below and focus on the entity information flow (authors can refer to (4) for the word information flow).

a) Entity Sequence Encoder. Suppose the *i*th sentence $s_i, i = 1, 2, \dots, L$ in a document contains M types of entities (e.g., gene entity type and chemical entity type), we denote e_{it}^j as the *t*th ($t = 1, 2, \dots, N$) entity of entity type j ($j = 1,2, \dots, M$) in the sentence s_i. The entities are embedded into vectors using a pre-trained embedding matrix E, which results in the entity representation $\tilde{e}_{it}^j = Ee_{it}^j$.

b) Bidirectional GRU. We use a bidirectional GRU (6) to incorporate the contextual information from both directions of entities, i.e., a forward GRU reads the input sequence as it is ordered from e_{i1}^j to e_{iN}^j and a backward GRU reads the sequence in the reverse order from e_{iN}^j to e_{i1}^j. The forward GRU results in a sequence of forward hidden states $(\vec{h}_{i1}^j, \vec{h}_{i2}^j, \dots, \vec{h}_{iT}^j)$ by the following equations:

$$\vec{h}_{it}^j = \left(1 - \vec{z}_{it}^j\right) \circ \vec{h}_{i\,t-1}^j + \vec{z}_{it}^j \circ \tilde{\vec{h}}_{i\,t-1}^j,$$

where

$$\tilde{\vec{h}}_{i\,t-1}^j = tanh(W^j \tilde{e}_{it}^j + U^j(\vec{r}_{it}^j \circ \vec{h}_{i\,t-1}^j))$$
$$\vec{z}_{it}^j = \sigma(W_z^j \tilde{e}_{it}^j + U_z^j \vec{h}_{i\,t-1}^j)$$
$$\vec{r}_{it} = \sigma\left(W_r^j \tilde{e}_{it}^j + U_r^j \vec{h}_{i\,t-1}^j\right),$$

and $W^j, W_z^j, W_r^j, U^j, U_z^j, U_r^j$ are weight matrices associated with entity type j, $\sigma(\cdot)$ is a logistic sigmoid function, and $\vec{z}_{it}^j$ and $\vec{r}_{it}^j$ are update gate and reset gate, respectively. Similarly, the backward GRU results in a sequence of backward hidden states $(\overleftarrow{h}_{i1}^j, \overleftarrow{h}_{i2}^j, \dots, \overleftarrow{h}_{iT}^j)$ which can be computed similarly. By concatenating $\vec{h}_{it}^j$ and $\overleftarrow{h}_{it}^j$, we have an annotation denoted as h_{it}^j (i.e., $h_{it}^j = [\vec{h}_{it}; \overleftarrow{h}_{it}]$) for the entity e_{it}^j capturing both preceding entities and following entities.

c) Entity Level Attention Layer. Since the entities contribute differentially to understand the sentence, we use attention mechanism to let the model pay more attention to individual entities that are important for the classification task. We use a softmax function on the entity annotation h_{it}^j to obtain a normalized importance weight α_{it}^j :

$$\alpha_{it}^j = \frac{exp(u_{it}^{j\prime} \delta_e^j)}{\sum_t u_{it}^j \delta_e^j}$$

where $u_{it}^j = \tanh(W_u^j h_{it}^j + U_u^j)$ is a hidden representation of h_{it}^j and δ_e^j is a high level representation learned during training indicating the importance of entities. Then the entity-bunch b_i^j can be computed by $b_i^j = \sum_t \alpha_{it}^j h_{it}^j$ with the focus on important entities of entity type j from the sentence s_i.

d) Entity-Bunch Encoder. Since a document contains L entity bunches (i.e., L sentences), we use a bidirectional GRU to encode the entity-bunch b_i^j. The forward GRU is computed by:

$$\vec{h}_i^j = \left(1 - \vec{z}_i^j\right) \circ \vec{h}_{i-1}^j + \vec{z}_i^j \circ \tilde{\vec{h}}_{i-1}^j$$

and the back ward GRU $\overleftarrow{h}_i^j$ is computed similarly. We get an annotation for the entity-bunch by concatenating $\overrightarrow{h}_i^j$ and $\overleftarrow{h}_i^j$, i.e., $h_i^j = [\overrightarrow{h}_i^j; \overleftarrow{h}_i^j]$.

e) Sentence and Entity-Bunch Level Attention Layer. To pay more attention on the sentences and entity-bunches that are important to classify a document, we also leverage attention mechanism to take the sentences and entity-bunches as input simultaneously. By doing so, the model is able to focus more or less on a combination of words (i.e., sentences) or a combination of entities (i.e., entity-bunch). We feed the sentence annotations and entity-bunch annotations into an attention mechanism by

$$\alpha^j = \frac{exp(u_i^{j\prime}\delta)}{\sum_i u_i^{j\prime}\delta}$$

where $u_i^j = \tanh(W_u h_i^j + U_u)$. We can obtain a document representation by summing over all the entity-bunches and the sentence:

$$d = \sum_i^L \alpha_i h_i + \sum_j^M \sum_i^L \alpha_i^j h_i^j.$$

f) Classification Layer. We use a softmax function on the document representation d and obtain the label by

$$y = argmax_y\, p(y|d),$$

$$p(y|d) = \varphi(W_d d + U_d)$$

where y is the label for document d, $\varphi(\cdot)$ is a softmax function, and W_d and U_d are weight matrices.

B. Entity Recognition and Normalization

In the *Document Triage Task*, a named entity recognition tool we developed in our previous studies (5, 7) was utilized to extract the entities from the given texts. For the named entity normalization, we used TaggerOne (8), a trainable semi-Markov structured linear classifier for any entity types. We did joint training for the genes and chemicals using diverse corpus from prior BioCreative shared task data sets. We supplemented the annotations provided by TaggerOne with annotations from other third-party tools such as beCAS (9) and Reach (10). Our strategy was to build consensus across multiple named entity recognition tools for the genes and chemicals.

For the *Relation Extraction Task*, we also utilized the above tool but differently we extracted genes and mutations by additionally using the annotations of tmVar (11).

III. Experiments and Results

A. Document Triage Task

The organizers provided 4,082 annotated PubMed articles (title and abstract), of which 1,729 (42.4%) were relevant documents describing genetic mutations affecting PPIs and 2,353 (57.6%) were non-relevant (see Table I). A set of 1,500 PubMed articles (title and abstract) was given as testing data to be classified into *relevant* or *non-relevant* document.

TABLE I. Statistics of the Training Data of Document Triage Task

Document #	Relevant Docs #	Non-relevant Docs #
4,082	1,729	2,353

We merged the title and abstract into a document and extracted two types of entities, i.e., genes and chemicals, from the document. In addition, we also obtained MeSH Terms using MetaMap (12) for each PubMed article. Though MeSH Terms were not entities, they could be fed into the proposed model effortlessly. Thus, there were totally three types of entities (M=3) utilized in the EHANN.

B. Relation Extraction Task

In the *Relation Extraction Task*, the organizers provided 597 training PubMed articles (title and abstract), with annotated proteins (and the corresponding Gene Entrez IDs) and interacting protein pairs mentioned in the text that are affected by a genetic mutation. Table II lists the statistics of this training data. The same set of 1,500 testing PubMed articles is given to recognize proteins (i.e., genes) first and then classify each protein pair into *interacted* or *non-interacted* pair by gene mutations.

TABLE II. Statistics of the Training Data of Relation Extraction Task

Document #	Gene ID #	Unique Gene ID #	Relation #	Avg # of Gene IDs per Doc	Avg # of Relations per Doc
597	8,833	1,316	760	14.8	1.3

To prepare the training data for the EHANN, for each pair of genes (e.g., Gene A and B) extracted by our tool from the given texts, we created a document consisting of all single sentences that contained both genes and all the intervals between two genes if they were located in different sentences (including Gene A and B). By doing so, each pair of genes was associated with a document. In addition, since the task aims to detect the mapping of gene products functional regions through the identification and study mutation that affecting the stability and affinity of molecular interactions, which means the detection of PPIs changed by genetic mutations is a crucial step (PPIM relationship). To achieve this goal, we considered the gene pairs between which one or more mutations occurred. Then we labeled the document as 1 if the relation of the corresponding gene pair was found in the given annotated data, and 0 otherwise.

C. Model Parameters

For both tasks, we evaluated the proposed model and tuned the hyperparamters on the training data using 10-fold cross-validation with random shuffling. We pre-trained the word embedding E on 1.25 million PubMed open access articles retrieved in March 2016. The dimension of E is set to 100. We set the GRU dimension to be 100. In each training process, we set epoch to 30 and batch size to 50. For training the model, we use categorical cross entropy as the loss function and RMSprop optimizer as the rate learning method.

D. Results

For the *Relation Extraction Task*, Table III shows the results of the named entity recognition. We extracted a total of 8,808 gene IDs from the training data, where there were 1,276 unique gene IDs. Note that those gene IDs are not necessarily occurred in the given annotated data. The precision, recall and F-measure for the gene entity recognition are 0.644, 0.642, and 0.643, respectively. Using the genes extracted by our tool, we created 9,047 pairs of genes. Among these pairs, 1,262 pairs were annotated as interacting pairs in the provided training data, so we assigned label '1' to these pairs and '0' to the rest of pairs. By doing so, we built a set of training data for the proposed model.

TABLE III. RESULTS OF NAMED ENTITY RECOGNITION FOR THE RELATION EXTRACTION TASK

# of Extracted Gene IDs	# of Extracted Unique Gene IDs	# of Gene Pairs	# of Gene Pairs with Relation	# of Gene Pairs without Relation
8,808	1,276	9,047	1,262	7,785

Table IV shows the classification results of the EHANN model for both tasks. The measurement is the average accuracy of 10-fold cross-validation on the training data. We achieved average accuracy of 96.65% (standard deviation is 9.57%) for the *Document Triage Task* and slightly lower average accuracy of 96.31% (standard deviation is 1.43%) for the *Relation Extraction Task.*

TABLE IV. EXPERIMENTAL RESULTS OF THE EHANN MODEL ON THE TRAINING DATA

Task	Average Accuracy (Std.)
Document Triage Task	96.65% (+/- 9.57%)
Relation Extraction Task	96.31 %(+/- 1.43%)

IV. DISCUSSION

As shown in Table IV, the performance of the proposed EHANN model is promising for both task based on the evaluation on the training data. We also observe that the standard deviation for the *Document Triage Task* is higher than the *Relation Extraction Task*. The reason might be that the data size of the former task is smaller than the latter task. For example, there are 3.6k documents in each round of the 10-fold cross validation used for training in the former task while there are 8.1k documents in the latter task.

A limitation in this work is the gene entity recognition tool used for the *Relation Extraction Task*. Since the precision and recall of extracting gene entities are low, the actual classification accuracy might drop significantly. Therefore, we will focus on gene entity recognition in the future work.

V. CONCLUSION

This paper describes the methods utilized in our participation of two tasks in the BioCreative VI PM track. We propose an Entity-enhanced Hierarchical Attention Neural Network (EHANN), which leverages three attention mechanisms to enable the model focus more on the important words, entities and sentences for the tasks. The experimental results show that the proposed EHANN model is effective for mining protein interactions from biomedical texts.

ACKNOWLEDGMENT

This work was made possible by NIGMS R01GM102282 and NLM R01LM11934.

REFERENCES

1. Collins FS, Varmus H. A new initiative on precision medicine. New England Journal of Medicine. 2015;372(9):793-5.
2. Singhal A, Simmons M, Lu Z. Text mining genotype-phenotype relationships from biomedical literature for database curation and precision medicine. PLoS computational biology. 2016;12(11):e1005017.
3. Oti M, Snel B, Huynen MA, Brunner HG. Predicting disease genes using protein–protein interactions. Journal of medical genetics. 2006;43(8):691-8.
4. Yang Z, Yang D, Dyer C, He X, Smola AJ, Hovy EH, editors. Hierarchical Attention Networks for Document Classification. HLT-NAACL; 2016.
5. Ravikumar K, Rastegar-Mojarad M, Liu H. BELMiner: adapting a rule-based relation extraction system to extract biological expression language statements from bio-medical literature evidence sentences. Database. 2017;2017(1):baw156.
6. Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:14090473. 2014.
7. Ravikumar K, Rastegar-Mojarad M, Liu H, editors. BELMiner – Information extraction system to extract BEL relationships. BioCreative Workshop VI; 2017 October 18-20; Bethesda, Maryland US.
8. Leaman R, Lu Z. TaggerOne: joint named entity recognition and normalization with semi-Markov Models. Bioinformatics. 2016;32(18):2839-46.
9. Nunes T, Campos D, Matos S, Oliveira JL. BeCAS: biomedical concept recognition services and visualization. Bioinformatics. 2013;29(15):1915-6.
10. Hahn MAV-EG, Surdeanu PTHM. A domain-independent rule-based framework for event extraction. ACL-IJCNLP 2015. 2015;127.
11. Wei C-H, Kao H-Y, Lu Z. PubTator: a web-based text mining tool for assisting biocuration. Nucleic acids research. 2013;41(W1):W518-W22.
12. Aronson AR, Lang F-M. An overview of MetaMap: historical perspective and recent advances. Journal of the American Medical Informatics Association. 2010;17(3):229-36.

Mining protein interactions affected by mutations using a NLP based machine learning approach

Jinchan Qu[1], Albert Steppi[1], Jie Hao[1], Jian Wang[1], Pei-Yau Lung[1], Tingting Zhao[2], Zhe He[3], Jinfeng Zhang[1,*]
[1]Department of Statistics, Florida State University, Tallahassee, FL 32306, USA
[2]Department of Geography, Florida State University, Tallahassee, FL 32306, USA
[3]College of Communication and Information, Florida State University, Tallahassee, FL 32306, USA

***Abstract*— The knowledge on the protein/gene interactions that are affected by mutations helps understand phenotype-genotype associations and predict disease prognosis and responses to treatments. Such information is scattered around in scientific literature and its manual curation is very time and resource consuming. Although much research has been done in the past to extract protein-protein interaction (PPI) information automatically from literature, much less has been done on extracting PPIs affected by mutations. An important step towards extracting such information is automatically retrieving the relevant articles. In this study, we classify abstracts as relevant or irrelevant, where being relevant means the article contains information that at least one PPI is affected by a mutation. We started out with a bag-of-words and bigrams model, and then added some frequency statistics. Interactions are modeled by triplets, where a triplet is defined as two protein names and an interaction word in a sentence. The Stanford Dependency Parser, which helps find the dependency graph for each sentence, was used to further calculate the shortest path distances for each of the three pairs of words in a triplet. We also used a model to estimate the probabilities that a triplet forms a true interaction. The shortest paths between protein name and mutation word pairs in the sentences were computed as well. Features based on all these above were used in a Gradient Boosting Trees model. Our model achieved satisfactory performance in the training data of the Biocreative VI challenge.**

Keywords*— *Protein-protein interactions; mutations; text mining; biomedical literature retrieval; protein interactions affected by mutations

I. Introduction

The publications in biomedical literature have been increasing at an accelerated speed. Reading all the literature in a particular domain of interest has become less and less feasible in recent years. And the gap between the number of papers a researcher can read and the number of all the papers he/she needs to read will only become wider over time. Searching literature using keywords at literature databases such as PubMed has been a very popular approach for finding relevant scientific articles. However, such searches can often return hundreds or even thousands of papers, many of which are not highly relevant to the topics a user aims to find. Biomedical literature triage (or retrieval) has been a popular topic in recent years [1-12]. Most of the methods use machine learning methods to retrieve relevant articles related to a particular topic. Literature triage tasks vary widely depending on the subdomain of the literature and also on the particular information one aims to retrieve.

In this study, we address the problem raised in the Biocreative challenge VI track 4, mining protein interactions and mutations for precision medicine. This challenge consists of two subtasks: (1) Document triage: identify relevant PubMed citations describing genetic mutations affecting protein-protein interactions; and (2) Relation extraction: extract experimentally verified protein-protein interactions (PPI) affected by the presence of a genetic mutation. Much research has been done in the past to extract protein-protein interaction (PPI) information automatically from literature [13-24]. Extracting PPIs affected by mutations is very challenging in that not only one needs to deal with information on PPIs, but also on whether they are affected by mutations. The information on whether a PPI is affected by mutations often occurs in sentences different from the one where PPIs are mentioned, which makes the tasks more challenging.

For document triage task, the training dataset consists of a set of ~4K PubMed articles. These articles are manually labelled as relevant/not relevant by BioGRID database [25] curators. The goal is to build automatic methods capable of receiving a list of PMIDs and return a relevance-ranked judgement of the test set for triage purposes.

For relation extraction task, a subset of the relevant articles in document triage task was manually annotated with relevant interacting protein pairs. Each PubMed article in this set has at least one interacting pair which is listed with the Gene Entrez ID of the two interactors. These protein-protein interactions have been experimentally verified and the analysis of natural occurring or synthetic mutations has identified protein residues crucial for the interaction. The goal is to build automated methods that are capable of receiving a set of PMID documents and return the set of interacting protein pairs (and their corresponding Gene Entrez IDs) mentioned in the text that are affected by a genetic mutation.

The validity of the text mining methods is evaluated using standard metrics such as average precision, f-measure, etc. In addition, we also used Area Under the Curve (AUC) to evaluate different models.

In the rest of the paper, we will first describe the method we developed for both tasks. We will then present our result on mainly the first task. We end the paper with conclusion and discussion.

This study was partially supported by a grant from NIGMS of National Institute of Health under award number R01GM126558.

II. METHODS

A. Data

The data used in this work are from PubMed articles. The dataset consists of titles and abstracts from 4082 papers, each manually labeled by BioGRID database curators as relevant or not relevant, for the training data, and 1500 papers for the test data.

B. Manual curation of mutation related words

Mutation related words were first manually curated from a list of words with a high TF-IDF between the true and false labels. When using cross validation to tune the model, mutation words not included in the training fold were excluded to avoid a potential information leak.

C. Feature extraction and model building

1) Triplet based modeling for protein-protein interactions

Protein-protein interactions (PPI) were modeled based on a triplet concept we developed in a previous study [26], where a triplet is defined as two protein names and an interaction word describing their interaction or relationship in the same sentence. The dictionary of interaction words were developed by a combination of manual and computational approach [26]. The method was later successfully used in knowledge discovery and other applications [27-30].

2) Bag-of-words and bigrams model

We started out with a Bag-of-words and bigrams model, and then added some frequency statistics from the title and body parts of the abstract, given that the title and the body have different information for triage. Treating the whole document as a bag, a word's potential to become a feature for training classifiers will depend on its term frequency in each abstract compared with its occurrence through the whole document. This way the words that appear often in the whole bag will be weighted less when deciding features. Besides the features from Bag-of-words, some frequency statistics were proved to be contributive in our initial Support Vector Machine model. These include: the percentage counts of protein/gene names, mutation words and interaction words, the percentages of sentences in an abstract that have 1, 2 or 3 and above pairs of protein name and mutation word within the sentence, the percentage count of within-sentence protein name and mutation word pairs in an abstract, and an indicator of whether a protein name and mutation word both appear in the title. Table 1 shows some of the most frequent words identified and kept in the model at this step.

We took this linear support vector machine (SVM) as our baseline model. It achieved an ROC-AUC score of 0.724 (std 0.0812) an F1 score of 0.5999 (std 0.05073). This linear SVM with bag-of-words features gives a baseline for our models performance.

TABLE I. MOST FREQUENT VOCABULARY AFTER BAG-OF-WORDS

word	count	word	count	word	count
protein/proteins	552	receptor	175	mutant/mutants	199
binding/binds/bind/binding to	507	signaling	156	show	132
domain/domains	351	dependent	152	membrane	129
kinase	235	dna	150	mutation/mutations	230
interaction/interactions/interacts /interact/interacting	564	human	145	induced	127
cell/cells	447	mediated	144	factor	125
phosphorylation	226	function/functional/functions	266	associated/association	187
complex/complexes	270	expression	138	have/has	219
activity/activity of	258	between	135	role	120
activation/activated	242	terminal	134	show that	119

3) Stanford Dependency Parser

The Stanford Dependency Parser was used to find the dependency graph for each sentence. For each triplet consisting of two protein names and an interaction word in a sentence, we computed the shortest path in the dependency graph between every pair of words in the triplet. Imagine each word is a node, each typed dependency is an edge that connects two nodes, then the parser can give the network graph of a sentence. Given two nodes, for example, Protein 1 and Protein 2, or Protein 1 and its interaction word with Protein 2, we need to find which path that connects these two nodes are the shortest. We can then obtained the Abstract-level features based on this information. For instance, the percentage counts of the triplets in an abstract that have the shortest path distances equaling 1, 2, 3, falling in the range 4 to 6, falling in the range 7 to 11, or equaling 12 and above, the percentage of sentences that have shortest path information in an abstract, etc.

We also used a model developed by another member of our group to extract true PPIs from the abstracts [26, 28, 31]. This model also uses information of the dependency graph generated by the Stanford Dependency Parser. Intuitively the shortest path features give us information to distinguish true PPIs among all the triplets. Triplets consisting of two proteins and an interaction word in the same sentence were then extracted from the abstracts and the model was used to estimate the probabilities that they form a true interaction. Therefore, by estimating the probabilities for triplets in an abstract to be true (PPIs), we added more features such as the percentage counts of the triplets with certain probability ranges. (We took the three equally sized bins [0, 1/3) (1/3, 2/3], (2/3,1].) We would then count the frequency of triplets with predicted probability falling into each bin.

The previous features give an idea of how many protein, interaction, and mutation words are contained in an abstract and an estimate for how many true PPIs are contained, but we have not yet done anything that connects the mutations to PPIs.

We computed the shortest paths in the dependency graphs between protein names and mutation words and added additional features based on these. For each protein word mutation word pair, we computed the shortest path between them. For each such pair, we computed the maximum probability given by our PPI extraction method over all triples containing the given protein. The probabilities were again put into bins [0 <= p < 1/3], [1/3 <= p < 2/3], [2/3 <= p <=1]. Shortest path lengths were again placed into the bins 1, 2, 3, 4-6, 7-11, 12 and above as before, then binned again according to

max PPI probability. For example, suppose there is a path connecting the protein tubulin with the mutation words SNP of length 8. This path will fit into the bin [7 <= sp <=11]. If our PPI extraction model gives a maximum probability of 0.56 that a triplet containing tubulin is a true PPI then this path will go into a bin with [7 <= sp <= 11] and [1/3 < p <= 2/3] since there are three probability bins and 6 shortest path length bins, there are a total of 18 bins a path length can go into. For each abstract, we take as a feature, the frequency of shortest paths lengths in each bin. For convenience we will call the probability bins: [0 <= p < 1/3] as bin 0, [1/3 < p < 2/3] as bin 1, [2/3 <= p <= 1] as bin 2.

We also employ a bag of words and bigrams approach to the words along each shortest path and to the dependencies along each path. If a path has a max PPI probability falling into bin 2, we append 2 to the end of each word in the path as follows: Tyorosines___2 recognized___2 trapped___2.

And similarly for dependencies. This shortest path had a max PPI probability falling into bin 1, so we have appended a 1 to each value. This is done so we can distinguish between paths connecting mutation words to proteins that have interactions from paths connecting to proteins that don't have any listed interactions: Xcomp___1 dep___1 xcomp___1 acl:relcl___1 root___1 dep___1 advmod___1 acl___1.

Frequencies of protein words, interaction words, and mutation words were also computed along the shortest paths.

4) Gradient boosting trees model

We tested three classifier types, a Linear SVM, a Random Forest, and a Gradient Boosted Trees model. The first two implemented in Python's scikit-learn. We used the popular XGBoost library[32] for gradient boosted trees. The number of words included in the bag of words and bigrams features was limited in the XGBoost model to avoid overfitting. The models were tuned using ROC-AUC scoring. The probability scores for the best model was then calibrated using Platt scaling[33]. Then cross validation was used to find a cutoff threshold to maximize the F1 score. A cutoff of p = 0.35 was chosen to predict abstracts as relevant. The model's performance on the training data was reasonable, with a cross-validated ROC-AUC score of 0.793 (std 0.018) and F1 score of 0.709 (std 0.0090).

III. Results

The following results were computed using ten-fold cross validation. Our baseline Linear SVM model using bag of words features and some count statistics received an ROC_AUC score of 0.74276. It was found that the SVM's Performance was sensitive to the choice of transformer weights in the scikit-learn FeatureUnion transformer. These weights were tuned with cross validation. We then enriched the feature set with information from the Stanford Dependency Parser and a PPI extraction model developed in-house based on some previous studies of ours [26, 28, 31]. We tried both a linear SVM and a Random Forest to the data with these additional features. With the additional features the ROC_AUC score of the linear SVM changed very slightly to 0.7428. A random forest with 1000 estimators, min samples leaf set at 2 and a Gini splitting criterion (these parameter values were found through cross-validation) achieved an ROC-AUC score of 0.78825 with these features, giving significantly better results than the SVM. Applying a tuned XGBoost to this same set of features yielded an ROC_AUC score of 0.78686. With all features included, the best scoring model was a tuned XGBoost with an ROC_AUC score of 0.79304. In this case, the performance of the Random Forest barely improved with the addition of the new features.

When tested on test data set, our method gave an F-1 score of 0.6778, recall of 0.8877, a precision of 0.5482. We ranked the fourth among all the participating teams based on F-1 score.

IV. Conclusions and Discussions

In this study, we tackled the problem of retrieving abstracts which contains at least one protein-protein interactions (PPI) affected by a mutation. We developed natural language processing (NLP) based methods, extracted a set of diverse features, and experimented with several popular machine learning methods. The performance on the training data of Biocreative VI was quite satisfactory.

There is still much room for further improvement of our models, especially in terms of optimizing the set of features we will use for the machine learning models. These will be the subject of future studies.

Acknowledgment *(Heading 5)*

We thank the help from Yuhang Liu, Zihan Cui, and Linlin Sha for reading some of the text and interpretation of the prediction results.

References

1. Wang, J.Z., et al., *G-Bean: an ontology-graph based web tool for biomedical literature retrieval.* BMC Bioinformatics, 2014. **15**.
2. Vishnyakova, D., et al., *Utilization of Ontology Look-Up Services in Information Retrieval for Biomedical Literature*, in *Data and Knowledge for Medical Decision Support*, B. Blobel, A. Hasman, and J. Zvarova, Editors. 2013. p. 155-159.
3. Cherdioui, S. and F. Boubekeur, *Information Retrieval Techniques for Knowledge Discovery in Biomedical Literature*. 2013 11th International Symposium on Programming and Systems, ed. A. Guessoum, Y. Djouadi, and A. Djouama. 2013. 137-142.
4. Dogan, R.I. and L. Yeganova, *Topics in machine learning for biomedical literature analysis and text retrieval.* BMC Bioinformatics, 2011. **12**.
5. Zhou, X.H., et al., *Relation-based document retrieval for biomedical literature databases*, in *Database Systems for Advanced Applications, Proceedings*, M.L. Lee, K.L. Tan, and V. Wuwongse, Editors. 2006. p. 689-701.

6. Singhal, A., M. Simmons, and Z. Lu, *Text Mining Genotype-Phenotype Relationships from Biomedical Literature for Database Curation and Precision Medicine.* PLoS Comput Biol, 2016. **12**(11): p. e1005017.
7. Kim, S., et al., *BioCreative V BioC track overview: collaborative biocurator assistant task for BioGRID.* Database (Oxford), 2016. **2016**.
8. Wei, C.H., et al., *tmVar: a text mining approach for extracting sequence variants in biomedical literature.* Bioinformatics, 2013. **29**(11): p. 1433-9.
9. Huang, C.C. and Z. Lu, *Community challenges in biomedical text mining over 10 years: success, failure and the future.* Brief Bioinform, 2016. **17**(1): p. 132-44.
10. Comeau, D.C., et al., *BioC: a minimalist approach to interoperability for biomedical text processing.* Database (Oxford), 2013. **2013**: p. bat064.
11. Wei, C.H., H.Y. Kao, and Z. Lu, *PubTator: a web-based text mining tool for assisting biocuration.* Nucleic Acids Res, 2013. **41**(Web Server issue): p. W518-22.
12. Wei, C.H., R. Leaman, and Z. Lu, *Beyond accuracy: creating interoperable and scalable text-mining web services.* Bioinformatics, 2016. **32**(12): p. 1907-10.
13. Bui, Q.-C., S. Katrenko, and P.M.A. Sloot, *A hybrid approach to extract protein-protein interactions.* Bioinformatics (Oxford, England), 2011. **27**: p. 259-265.
14. Bui, Q.-C., et al., *Extracting causal relations on HIV drug resistance from literature.* BMC bioinformatics, 2010. **11**: p. 101.
15. Ceol, A., et al., *Linking entries in protein interaction database to structured text: the FEBS Letters experiment.* FEBS letters, 2008. **582**: p. 1171-1177.
16. Chowdhary, R., J. Zhang, and J.S. Liu, *Bayesian inference of protein-protein interactions from biological literature.* Bioinformatics (Oxford, England), 2009. **25**: p. 1536-1542.
17. Giles, C.B. and J.D. Wren, *Large-scale directional relationship extraction and resolution.* BMC bioinformatics, 2008. **9 Suppl 9**: p. S11.
18. Hu, X. and D.D. Wu, *Data mining and predictive modeling of biomolecular network from biomedical literature databases.* IEEE/ACM transactions on computational biology and bioinformatics / IEEE, ACM, 2007. **4**: p. 251-263.
19. Huang, M., et al., *Mining physical protein-protein interactions from the literature.* Genome Biol, 2008. **9 Suppl 2**: p. S12.
20. Krallinger, M., et al., *Overview of the protein-protein interaction annotation extraction task of BioCreative II.* Genome biology, 2008. **9 Suppl 2**: p. S4.
21. Krallinger, M., F. Leitner, and A. Valencia. *Assessment of the {S}econd {B}io{C}reative {PPI} Task: {A}utomatic Extraction of Protein-Protein Interactions*. in *Proceedings of the Second BioCreative Challenge Evaluation Workshop*. 2007.
22. Pyysalo, S., et al., *Comparative analysis of five protein-protein interaction corpora.* BMC bioinformatics, 2008. **9 Suppl 3**: p. S6.
23. Tikk, D., et al., *A Comprehensive Benchmark of Kernel Methods to Extract Protein-Protein Interactions from Literature.* PLoS Computational Biology, 2010. **6**(7): p. e1000837.
24. Krallinger, M., et al., *The Protein-Protein Interaction tasks of BioCreative III: classification/ranking of articles and linking bio-ontology concepts to full text.* BMC Bioinformatics, 2011. **12 Suppl 8**: p. S3.
25. Chatr-Aryamontri, A., et al., *The BioGRID interaction database: 2017 update.* Nucleic Acids Res, 2017. **45**(D1): p. D369-D379.
26. Chowdhary, R., J. Zhang, and J.S. Liu, *Bayesian inference of protein-protein interactions from biological literature.* Bioinformatics, 2009. **25**(12): p. 1536-42.
27. Lindsey Bell, et al., *Integrated bio-entity network: a system for biological knowledge discovery.* PLoS One, 2011. **6**(6): p. e21474.
28. Bell, L., J. Zhang, and X. Niu, *Mixture of logistic models and an ensemble approach for extracting protein-protein interactions.* ACM-BCB, 2011: p. 371-375.
29. Chowdhary, R., et al., *Context-specific protein network miner--an online system for exploring context-specific protein interaction networks from the literature.* PLoS One, 2012. **7**(4): p. e34480.
30. Balaji, S., et al., *IMID: integrated molecular interaction database.* Bioinformatics, 2012. **28**(5): p. 747-9.
31. Zhang, J., *Automatic extraction of bio-entity relationships from literature. USPTO No. 8,886,522* 2014, Florida State University: USA.
32. Chen, T. and C. Guestrin, *XGBoost: A Scalable Tree Boosting System*, in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2016, ACM: San Francisco, California, USA. p. 785-794.
33. Platt, J.C., *Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods*, in *ADVANCES IN LARGE MARGIN CLASSIFIERS*. 1999, MIT Press. p. 61-74.

WBI@BioCreative
Track 4: Mining protein interactions and mutations for precision medicine (PM)

Jurica Ševa, Patrick Jähnichen, Ulf Leser
Knowledge Management in Bioinformatics, Department of Computer Science, Humboldt University of Berlin, Berlin, Germany

***Abstract*—Precision Medicine has recently been attracting increased interest from the scientific community. This is to be attributed both to availability of gold standard data sets as well as advances in text mining. We approached a task as a three step process, each dealing with one of the IE problems: document triage/classification, named entity recognition and relationship extraction as a joint task and finally named entity normalization. Our work relies on a combination of rule based dictionary/string matching, machine learning as well as neural networks.**

Keywords—protein-protein interaction, named entity recognition, named entity normalization, document classification, precision medicine

I. Introduction (*Heading 1*)

Recent advances in the automation of machine based analysis of written human language has led to an increased (research) interest in developing practical applications in a wide range of application domains. *Biomedical Natural Language Processing* (BioNLP) is one of the application domains where the impact of this automation has yielded a fast progress as well as offered even more complex pipelines to be built. The purpose of these pipelines is to assist specialists in resolving actual issues/diagnoses, by minimizing or even eliminating the need for manual effort.

Precision Medicine (PM) is a recent field, attracting a significant amount of attention from the research community. PM focuses on using genetic profiles of individuals in developing treatments by analyzing links between genes, mutations, and diseases with (specialized) treatments which are available in scientific/research literature (1). As such, it presents a novel task for the BioNLP community. To be able to provide proper PM services to working physicians, and extract complex information from unstructured text, there is a growing need to develop and implement automation pipelines which look for various entities, which were changed, or manipulated in any other way, by various factors. One such challenge is presented with the BioCreative VI Track 4[1]. The focus of this years track is the identification of protein-protein interactions but with a single distinction; the focus is extended to finding such entities where the proteins were affected by genetic mutations. This promises to give a more detailed view in the patients medical profile, and increase chances for a successful diagnosis and treatment.

As presented, this task is divided in two parts, each dealing with one of the fundamental text mining (TM) tasks in NLP: *document classification/triage* and *relation extraction (RE)*. Furthermore, to solve the second part of the challenge one needs to solve two additional problems, namely *named entity recognition (NER)* (with regards to genes/proteins mentioned in the text) and *named entity normalization (NEN),* against NCBI Gene Database. In solving these tasks, and combining them into an automated system, we used several approaches. For NER and RE we utilized neural networks, for document triage we used a standard supervised machine learning approach and for NEN we relied on string/dictionary matching, backed up with using available online resources.

This paper is structured as follows: next, we give an overview of related work, with regard to the 4 recognized tasks. This is followed by a detailed presentation of our approach, for each of the relevant tasks, with obtained results based on the training data (where available). This is followed by a discussion of encountered problems as well as (possible) future work.

II. Related work

Text mining has a long tradition in the filed of BioNLP. It was used to solve a wide range of tasks, thus increasing the level of possible automation in this application domain. There has been a number of extensive work done with information related to cancer (a good overview of the field is presented in (1)–(12)). Machine learning have shown to produce state of the art results for a wide range of NLP tasks. Recently, as with other NLP tasks, the focus has shifted toward neural networks (NN), as in (13). Additionally, the use of word embeddings, as feature representation, has greatly helped in improving upon current state-of-the-art results.

Document triage/classification is one of the oldest tasks in NLP. Recent work has shown that the best performing approach largely depends on the type of text in the corpus (e.g. tweets vs. full text documents). Recent advances in this task focus on the use of word embeddings, such es word2vec (14), GloVe (15), FastText (16) or Starspace (17) which also greatly improves achieved results. While word2vec and GloVe represent state-of-the-art representations, where each token is represented by an N

[1]http://www.biocreative.org/tasks/biocreative-vi/track-4/

dimensional feature vector instead (and thus using the information about the context of the token as well as the token), FastText and Starspace offer the possibility to simultaneously learn word embeddings and solve one or more of NLP tasks through their interface (e.g. document classification).

Both NER and RE can be seen as subtasks of information extraction (18). While early approaches rely on knowledge-based approaches making use of external ontologies like WordNet[2], recent advances in research on artificial neural networks have led to a variety of methods. Current state of the art, in various (Bio)NLP tasks is the use of bi-directional Long Short Term Memory (LSTM) with word or character embeddings. There is a large corpora of research articles where this combination outperforms other approaches (e.g (19)–(21)). Recently, the use of Convolutional Neural Network (CNN) architectures has show to be on par with LSTM, but offering a great improvement in the computational speed. Liu et. al (22) use a CNN to extract relations using additional semantic information via synonym dictionaries. Without using any external information (23) achieves state-of-the-art results, being among the first methods to utilize recurrent neural networks. Building on this, (24) models the tasks of NER and RE jointly, additionally introducing bidirectional recurrent layers to better exploit the structural features of the text. This is the model we use for our analysis. There also are various available tools which address, with a varying success rate, the problem of NER for various entities. For identifying genes from free text there are various tools: *GnormPlus* (25), *Neji* (26), *GNAT* (27) to name a few. Additionally, some of these tools perform entity normalization as well (to one of the possible standards). Another important field, which is of use in solving the task of the track 4, is finding mentions of mutations in free text. Genetic mutations represent changes in the human DNA and are result of errors which occurred during DNA replication. This has been researched extensively as well. Therefore there are tools which can be used as a black box. *tmVar* (28), *Nala* (5) and *SETH* (29) currently produce state-of-the-art results on available training corpora. NER and NEN are often seen an accompanying steps in extraction various entities from text.

Entity normalization is a common task in the BioNLP text mining pipeline, mostly due to a number of standards that coexist for the same entity (type). Entity normalization has been addressed by a number of work in scientific community. To solve this task one usually has to approach it as a word sense disambiguation (WSD) problem, as there usually exist several possible candidates to choose the identifier from. A valuable resource for this task was presented through the BioCreative II gene normalization task (30). As such, it represents a good starting point as it offers gold standard data sets. The task was repeated in the BioCreative III challenge (18). These efforts have resulted in an increased research effort as well as plethora of approaches and tools, of which we will mention a few. The majority of approaches still focus on the use of dictionaries, (various) knowledge sources and string matching for normalization. This has also inspired our work in this challenge. Another challenge in normalization is normalizing identifiers related to a specific species. For example, *PgenN* (32) is focused on genes normalization for plants, (33) extends normalization across species while *NTTMUNSW BioC* module (21) extends the normalization to full texts. *ProNormz* (35) also utilizes dictionaries "with of 15 string matching rules and a disambiguation module to achieve the normalization". Hakenberg et al. (36) approach the task "by using background knowledge on each gene, namely annotations related to function, location, disease, and so on". Various other tools also perform the same action, e.g. (37)–(39). An important resources is also represented through the *UniProt* interface (40).

III. METHODOLOGY

As presented before, our pipeline is defined through four tasks to be performed. Here a brief description of each of the steps in the pipeline is given.

A. Document Triage

As the first step in the pipeline, the focus of this task is on automatic classification of documents which carry relevant information for the task. We approached this step as a binary classification problem, with documents being either relevant or not relevant. Although the use of word embeddings, as mentioned in the previous section, usually offers state-of-the-art results, we did not make any use of them in this step. Our work relied on testing various machine learning algorithms to create the best performing model, where word embeddings are not useful. Instead, we heavily relied on feature engineering with automatic exploration and optimization of hyperparameter space of tested models.

To achieve best results, we have implemented a pipeline which trains and evaluates several machine learning models, using *word n-grams*, with *n* in range from (1,1) to (1,3). Additionally, we have explored two features spaces with respect how the used vocabulary has been created. In the first approach we selected features which were relative complements of features with respect to the two classes (*important, not important)*. In the second approach, we relied on an automatic creation of the feature space, with feature selection based on chi^2 test (with a cut off of 10% most important features). The results of this step are presented in Figure 1.

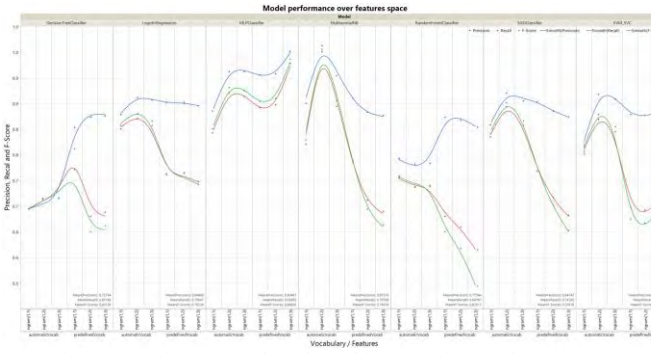


Figure 1: Performance of used Machine Learning models

The best performing model was *Multinomial Naive Bayes* with automatic vocabulary and 2-gram based feature space,

[2]https://wordnet.princeton.edu/

followed closely by Multi-layer Perceptron (MLP) based on manually created vocabulary and 3-gram based features space. The other tested models were *Decision Trees, Random Forests, Lasso, Stohastic Gradient Descent, Linear Support Vector Machine, Elastic Net, Support Vector Machine and Logistic Regression*

B. Named Entity Recognition and Relation Extraction

Both problems of NER and RE are handled jointly following the recent approach of (24). In general, we assume that relations appear on a sentence level. Hence, we apply a two stage process per sentence of a document that has been labeled as relevant. In the first stage, we identify all named entities that appear in the sentence. To do so, we first generate a set of features for each token in the sentence (a character representation based on character embeddings per token, a word embedding as learned from word2vec and the POS tag as learned from the Stanford NLP toolkit) and then feed a concatenated feature vector through a recurrent neural network. The architecture of choice here is a bidirectional LSTM layer, on top of which a fully connected activation layer is placed, and finally a softmax layer is used to generate the binary classification per token.

After having identified candidate named entities we extract the sequence between each pair of candidates, again generate a feature vector based on this sequence and feed this through another bidirectional LSTM layer. The parameters of both bidirectional layers are shared between the NER and RE tasks. In this way, both tasks can benefit from a type of transfer learning of related tasks. Finally, the LSTM layer output is again directed through a fully connected layer which is finally followed by a softmax layer for classifying a present or nonexistent relation between the candidates.

The described model has been implemented using the Python package *Keras* [3] and optimization is performed using the categorical cross entropy loss function and the Adam optimizer (41).

C. Named Entity Normalization

As the last step in the pipeline, NEN tries to find the appropriate Entrez gene ID for the input text. It heavily relies on the second step in the pipeline, NER, as the errors in that step propagate on to this stage. For NEN we have built a pipeline which utilizes several data sources and tries to normalize the input text, with respect to a possible species type, against the NCBI Entrez gene ID database. An overview of the pipeline has been given in Figure 2. The input to this part of the pipeline is a triple *(PubMed ID, Gene Name 1, Gene Name 2)*.

[3]https://keras.io/

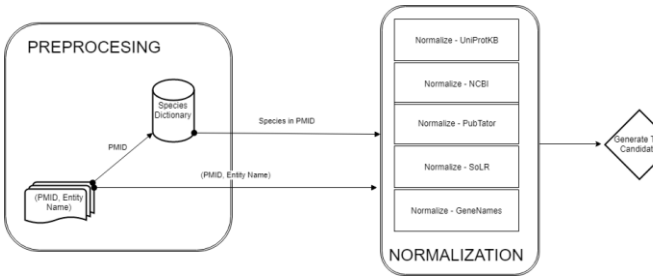


Figure 2: Named Entity Normalization process

Our pipeline utilizes a combination of dictionary lookup and string matching approach against multiple sources, both offline as well as online, which were available for use during the development. Each of the normalization sources returns none or more possible candidates, with the maximum limit set to 10 per source. After the candidates have been received from all sources, a count based heuristic is used to select the most suitable, single candidate. In case of multiple IDs with the same maximal count, the first identifier is used. As such, this approach performs admirably in case of a few possible candidates generated in the pipeline and underperforms when multiple valid candidates (e.g. with the same count value) have been identified. As expected, this part of the pipeline impacts the final results in a negative way. The results are shown in table 1.

Precision	Recall	F-Score
0.56	0.56	0.55

Table 1. Performance of Named Entity Normalization

IV. Discussion and (possible) future work

In this work we presented our pipeline developed for solving the task presented in the BioCreative VI track 4. As this represents a complex pipeline, which combines solution(s) to several mentioned problems in BioNLP, the performance is as good as the weakest link in the chain.

Our approach, and the achieved results (as evaluated on the available gold standard train data), can be viewed more as a strong baseline than a final solution. There are several aspects of our work which need additional focus. That is especially true for the last step in the pipeline, NEN, where a dictionary lookup and string matching approach has shown to be lacking, especially in the case when multiple valid candidates can be chosen as the final ID. One direction this can be taken to is the use of machine learning and, possibly, neural networks with word embedding feature representation.

Although our approach to solve the NER and RE tasks of the pipeline have proven satisfactory, this approach has also shown shortcomings. As the current approach is still a two-stage approach, we found it difficult to prevent errors made in the first stage to propagate to the second one. We are planning to incorporate regularization structures such as drop-out layers into the network to mitigate overfitting and error propagation. Another interesting step towards a truly joint model of named

entities and their relations would be to construct a network that does not learn entities and relations separately.

V. References

[1] Singhal, A., M. Simmons, and Z. Lu, "Text Mining Genotype-Phenotype Relationships from Biomedical Literature for Database Curation and Precision Medicine," *PLOS Comput. Biol.*, vol. 12, no. 11, p. e1005017, Nov. 2016.

[2] Rebholz-Schuhmann, D., A. Oellrich, and R. Hoehndorf, "Text-mining solutions for biomedical research: enabling integrative biology," *Nat. Rev. Genet.*, vol. 13, no. 12, pp. 829–839, Nov. 2012.

[3] Zhu, F. *et al.*, "Biomedical text mining and its applications in cancer research," *J. Biomed. Inform.*, vol. 46, no. 2, pp. 200–211, Apr. 2013.

[4] Hirschman, L. *et al.*, "Text mining for the biocuration workflow," *Database*, vol. 2012, no. 0, p. bas020-bas020, Apr. 2012.

[5] Cejuela, J. M. *et al.*, "nala: text mining natural language mutation mentions," *Bioinformatics*, vol. 33, no. 12, pp. 1852–1858, Jun. 2017.

[6] Wei, C.-H., B. R. Harris, H.-Y. Kao, and Z. Lu, "tmVar: a text mining approach for extracting sequence variants in biomedical literature," *Bioinformatics*, vol. 29, no. 11, pp. 1433–1439, Jun. 2013.

[7] Singhal, A., M. Simmons, and Z. Lu, "Text Mining Genotype-Phenotype Relationships from Biomedical Literature for Database Curation and Precision Medicine," *PLOS Comput. Biol.*, vol. 12, no. 11, p. e1005017, Nov. 2016.

[8] Spasić, I., J. Livsey, J. A. Keane, and G. Nenadić, "Text mining of cancer-related information: Review of current status and future directions," *Int. J. Med. Inform.*, vol. 83, no. 9, pp. 605–623, Sep. 2014.

[9] Czarnecki, J., I. Nobeli, A. M. Smith, and A. J. Shepherd, "A text-mining system for extracting metabolic reactions from full-text articles," *BMC Bioinformatics*, vol. 13, no. 1, p. 172, 2012.

[10] Simmons, M., A. Singhal, and Z. Lu, "Text Mining for Precision Medicine: Bringing Structure to EHRs and Biomedical Literature to Understand Genes and Health," in *Translational Biomedical Informatics - A Precision Medicine Perspective*, vol. 939, Shen, B., H. Tang, and X. Jiang, Eds. Singapore: Springer Singapore, 2016, pp. 139–166.

[11] Fleuren, W. W. M. and W. Alkema, "Application of text mining in the biomedical domain," *Methods*, vol. 74, pp. 97–106, Mar. 2015.

[12] Wei, C.-H., H.-Y. Kao, and Z. Lu, "PubTator: a web-based text mining tool for assisting biocuration," *Nucleic Acids Res.*, vol. 41, no. W1, pp. W518–W522, Jul. 2013.

[13] Baker, S., A. Korhonen, and S. Pyysalo, "Cancer Hallmark Text Classification Using Convolutional Neural Networks."

[14] Mikolov, T., G. Corrado, K. Chen, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," *Proc. Int. Conf. Learn. Represent. (ICLR 2013)*, vol. 1, pp. 1–12, 2013.

[15] Pennington, J., R. Socher, and C. Manning, "Glove: Global Vectors for Word Representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543.

[16] Bojanowski, P., E. Grave, A. Joulin, and T. Mikolov, "Enriching Word Vectors with Subword Information," 2016.

[17] Wu, L., A. Fisch, S. Chopra, K. Adams, A. Bordes, and J. Weston, "StarSpace: Embed All The Things!," Sep. 2017.

[18] Cowie, J. and W. Lehnert, "Information extraction," *Commun. ACM*, vol. 39, no. 1, pp. 80–91, Jan. 1996.

[19] Habibi, M., L. Weber, M. Neves, D. L. Wiegandt, and U. Leser, "Deep learning with word embeddings improves biomedical named entity recognition," *Bioinformatics*, vol. 33, no. 14, pp. i37–i48, Jul. 2017.

[20] Young, T., D. Hazarika, S. Poria, and E. Cambria, "Recent Trends in Deep Learning Based Natural Language Processing," 2017.

[21] Miwa, M. and M. Bansal, "End-to-End Relation Extraction using LSTMs on Sequences and Tree Structures," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016, pp. 1105–1116.

[22] Liu, C. Y., W. B. Sun, W. H. Chao, and W. X. Che, "Convolution neural network for relation extraction," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2013, vol. 8347 LNAI, no. PART 2, pp. 231–242.

[23] Zheng, S., J. Xu, P. Zhou, H. Bao, Z. Qi, and B. Xu, "A neural network framework for relation extraction: Learning entity semantic and relation pattern," *Knowledge-Based Syst.*, vol. 114, pp. 12–23, Dec. 2016.

[24] Li, F., M. Zhang, G. Fu, and D. Ji, "A neural joint model for entity and relation extraction from biomedical text," *BMC Bioinformatics*, vol. 18, 2017.

[25] Wei, C.-H., H.-Y. Kao, and Z. Lu, "GNormPlus: An Integrative Approach for Tagging Genes, Gene Families, and Protein Domains," *Biomed Res. Int.*, vol. 2015, pp. 1–7, 2015.

[26] Campos, D., S. Matos, and J. Oliveira, "Neji: a tool for heterogeneous biomedical concept identification," *Proc. BioLINK SIG*, vol. 20, no. 7, pp. 1178–1190, May 2013.

[27] Hakenberg, J. *et al.*, "The GNAT library for local and remote gene mention normalization," *Bioinformatics*, vol. 27, no. 19, pp. 2769–2771, Oct. 2011.

[28] Wei, C.-H., L. Phan, J. Feltz, R. Maiti, T. Hefferon, and Z. Lu, "tmVar 2.0: Integrating genomic variant information from literature with dbSNP and ClinVar for precision medicine," *Bioinformatics*, Sep. 2017.

[29] Thomas, P., T. Rocktäschel, J. Hakenberg, Y. Lichtblau, and U. Leser, "SETH detects and normalizes genetic variants in text," *Bioinformatics*, vol. 32, no. 18, pp. 2883–2885, Sep. 2016.

[30] Morgan, A. A. *et al.*, "Overview of BioCreative II gene normalization.," *Genome Biol.*, vol. 9 Suppl 2, no. Suppl 2, p. S3, 2008.

[31] Lu, Z. *et al.*, "The gene normalization task in BioCreative III," *BMC Bioinformatics*, vol. 12, no. Suppl 8, p. S2, 2011.

[32] Ding, R., C. N. Arighi, J.-Y. Lee, C. H. Wu, and K. Vijay-Shanker, "pGenN, a gene normalization tool for plant genes and proteins in scientific literature.," *PLoS One*, vol. 10, no. 8, p. e0135305, 2015.

[33] Wei, C.-H. and H.-Y. Kao, "Cross-species gene normalization by species inference," *BMC Bioinformatics*, vol. 12, no. Suppl 8, p. S5, Oct. 2011.

[34] Dai, H.-J., O. Singh, J. Jonnagaddala, and E. C.-Y. Su, "NTTMUNSW BioC modules for recognizing and normalizing species and gene/protein mentions.," *Database (Oxford).*, vol. 2016, 2016.

[35] Subramani, S., K. Raja, and J. Natarajan, "ProNormz – An integrated approach for human proteins and protein kinases normalization," *J. Biomed. Inform.*, vol. 47, pp. 131–138, Feb. 2014.

[36] Hakenberg, J., C. Plake, L. Royer, H. Strobelt, U. Leser, and M. Schroeder, "Gene mention normalization and interaction extraction with context models and sentence motifs," *Genome Biol.*, vol. 9, no. Suppl 2, p. S14, 2008.

[37] Hakenberg, J. *et al.*, "The GNAT library for local and remote gene mention normalization," *Bioinformatics*, vol. 27, no. 19, pp. 2769–2771, Oct. 2011.

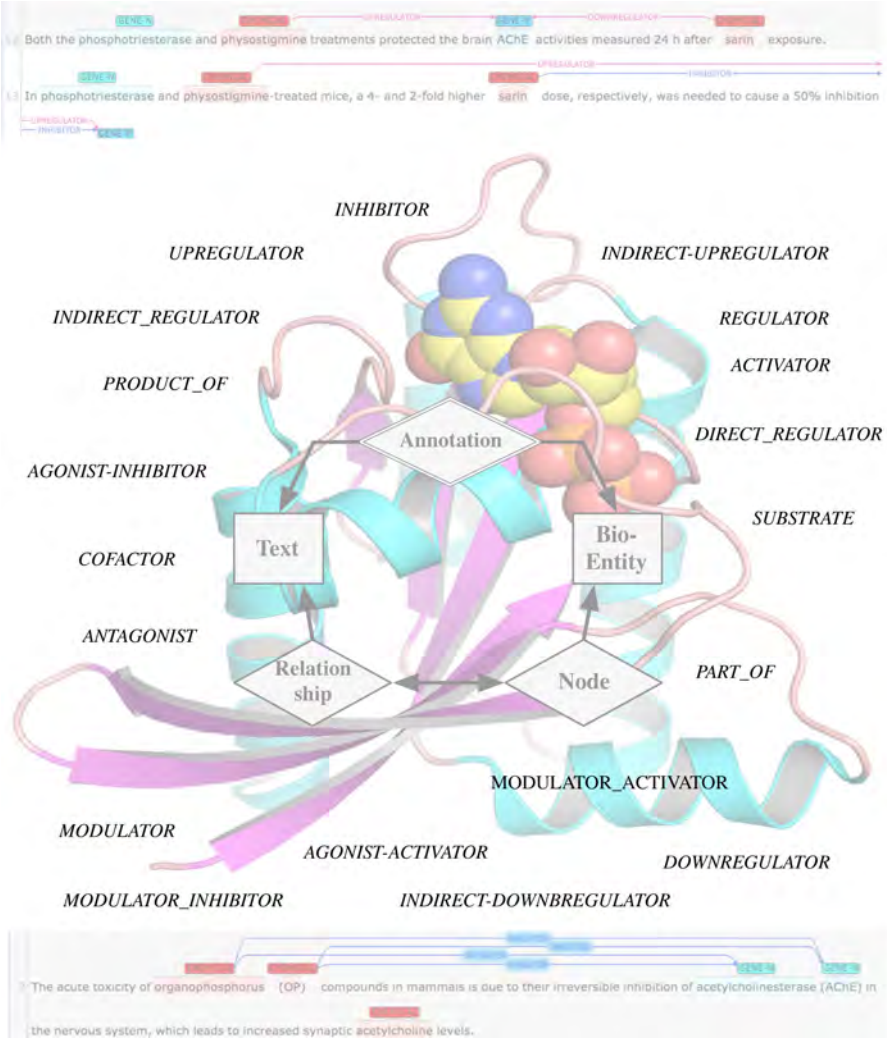
[38] Hu, Y., Y. Li, H. Lin, Z. Yang, and L. Cheng, "Integrating Various Resources for Gene Name Normalization," *PLoS One*, vol. 7, no. 9, p. e43558, Sep. 2012.

[39] Huang, M., J. Liu, and X. Zhu, "GeneTUKit: a software for document-level gene normalization.," *Bioinformatics*, vol. 27, no. 7, pp. 1032–3, Apr. 2011.

[40] UniProt Consortium, "UniProt: a hub for protein information," *Nucleic Acids Res.*, vol. 43, no. D1, pp. D204–D212, Jan. 2015.

[41] Kingma, D. P. and J. Ba, "Adam: A Method for Stochastic Optimization," pp. 1–15, Dec. 2014.

Track 5
Text mining chemical-protein interactions



Overview of the BioCreative VI chemical-protein interaction Track

Martin Krallinger[1§], Obdulia Rabal[2], Saber A. Akhondi[3], Martín Pérez Pérez[4], Jesús Santamaría[1], Gael Pérez Rodríguez[4], Georgios Tsatsaronis[3], Ander Intxaurrondo[1], José Antonio López[1] Umesh Nandal[3], Erin Van Buel[3], Akileshwari Chandrasekhar[3], Marleen Rodenburg[3], Astrid Laegreid[5], Marius Doornenbal[3], Julen Oyarzabal[2], Analia Lourenço[4], Alfonso Valencia[6]

[1]Spanish National Cancer Research Centre, Madrid, Spain
[2]Center for Applied Medical Research (CIMA), University of Navarra, Spain
[3]Content and Innovation, Elsevier BV, Netherlands
[4]University of Vigo, Spain
[5]Norwegian University of Science and Technology (NTNU), Norway
[6]Barcelona Supercomputing Center, Spain

§ Corresponding author

Abstract*—Despite the considerable number of available systems that recognize automatically mentions of genes/proteins and chemicals in text, only a limited number of attempts were made so far to extract interactions between them. Most biomedical relation extraction systems focus on the extraction of protein-protein or gene/chemical-disease relations. The detection of interactions between drugs and proteins/genes is of key relevance for pharmacological and clinical research, playing an important role for drug discovery, understanding of molecular mechanism of adverse drug reactions, describing drug metabolism or drawing regulatory networks of importance for systems pharmacology. The BioCreative VI - ChemProt track represents the first attempt to promote the development of systems for extracting chemical-protein interactions (CPIs), of relevance for precision medicine as well as for drug discovery and basic biomedical research. The novel ChemProt corpus consists of text exhaustively annotated by hand with mentions of chemical compounds/drugs and genes/proteins, as well as 22 different types of compound-protein relations. To focus on a subset of important relations, 5 relation classes were used for evaluation purposes, including agonist, antagonist, inhibitor, activator and substrate/product relations. A total of 13 participating teams returned 45 runs for this track. Despite the biological complexity of the considered relation types, top-scoring teams could obtain an F-measure across relation classes of 64.10%. Performance varied depending on the relation class: for the antagonist relation class the best team obtained an F-measure of 72.56% (precision of 80.75%, recall of 65.87%) while for inhibition/down-regulation the best value was of 71.48% (with a precision of 76.51% and a recall of 67.07%).

Keywords—text mining; chemical compound; drug; protein; drug target; agonist; antagonist, inhibitor; activator; gene regulation; chemical-protein relation

I. Introduction

A growing amount of scientific articles, medicinal chemistry patents and other biomedical documents provide descriptions of interactions between chemical compounds and gene products. Compared to the extraction of protein-protein (1) or gene/chemical-disease relations (2), the detection of associations between chemical entities (e.g. drugs or active pharmaceutical ingredients) and proteins/genes is an underexplored biomedical text mining research area.

There is an increasing interest in the integration of chemical and biomedical data understood as the curation of relationships between biological and chemical entities from text and storing such information in form of structured annotation databases. Such databases are of key relevance not only for biological but also for pharmacological and clinical research. Certain types of chemical-protein/gene interactions are of key relevance for biomedicine, including metabolic relations (e.g. substrates, products), antagonist, agonist, inhibitor or activator associations.

Despite the existence of competitive named entity recognition tools for tagging chemicals and genes/proteins, the retrieval of certain relationships between these two types of entities using text mining and information extraction approaches has only be attempted by a limited number of systems. A comprehensive review of previously published text mining systems for different types of chemical-protein relations is summarized in Krallinger et al. 2017 (3). For instance, Rindflesch et al. introduced an early system called EDGAR (4) that extracted drug-gene relations (drugs altering gene expression) and gene-drug relations (gene/protein altering drug activity) by exploiting syntactic information and relational vocabulary. The drug-target interaction resource SuperTarget (5) applied the EBIMed tool to select text passages potentially describing drug-target associations (6). Tari et al. applied syntactic dependencies to recover gene-drug relations (7) while Czarnecki et al. (8) published a pattern-matching and rule-based approach for detecting metabolic reaction relations. The LimTox online server incorporates both pattern matching and machine learning techniques to retrieve gene expression induction, inhibition and metabolism relations between cytochromes P-450 (CYPs) and drugs (9). The relation patterns used by LimTox, including also protein-protein interactions

patterns, were released as part of the LimTox resource collection. The reasons for missing publicly available text mining resources for chemical-protein interaction extraction lay mainly in: (a) the difficulty to define chemical-protein interactions in the first place, despite some attempts to define the underlying relation concepts, (b) the lack of large corpora of manually curated text bound annotations of mentions of chemical entities and genes/proteins and their interactions, done by domain experts and (c) the lack of publicly shared annotation guidelines and rules for labeling mentions of chemicals, genes and proteins relationships.

Due to the critical importance of CPI we have posed a track specially devoted to this kind of biological relation at the BioCreative VI community evaluation effort.

II. THE CHEMPROT TRACK AND CORPUS

The ChemProt track aimed to promote the development of systems able to extract chemical-protein interactions that are relevant for precision medicine, drug discovery and basic biomedical research. To carry out the ChemProt track it was necessary to create a novel manually annotated corpus that annotates exhaustively all the mentions of the underlying entity types, namely chemical compounds and drugs as well as genes and proteins. In order to make this feasible in practice, and to focus on types of text that are easily accessible, we restricted the annotation process to PubMed abstracts essentially published between 2005 and 2014. Domain experts with experience in text annotation and database curation tasks annotated by hand all abstracts. The manual labeling of chemicals and genes was done separately to avoid cross-influence during the annotation process. The labeling of mentions of chemical entities, genes and proteins was done following the annotation process and guidelines previously used for several BioCreative tasks (10–12). Gene/protein mentions were manually normalized to their corresponding database identifiers whenever possible and classified as either normalizable to databases (tag: GENE-Y) or non normalizable mentions (GENE-N). Participanting teams were only provided with this classification of gene mentions and not the actual database identifier to avoid usage of external knowledgebases for producing their predictions.

After the completion of chemical and gene annotations, all mention annotations were merged in order to carry out the relation annotation process. To enable the annotation of chemical-protein interactions, the ChemProt track organizers constructed very granular relation annotation rules described in a 33 pages annotation guidelines document. These guidelines were refined during an iterative process based on the annotation of sample documents.

The guidelines provided the basic details of the chemical-protein interaction annotation task and the conventions that had to be followed during the corpus construction process. They incorporated suggestions made by curators as well as observations of annotation inconsistencies encountered when comparing results from different human curators.

In brief, the annotated ChemProt interactions included direct interactions (when a physical contact exists between a chemical and a gene/protein) as well as indirect regulatory interactions that alter either the function or the quantity of the gene/gene product. The aim of the iterative manual annotation cycles was to improve the quality and consistency of the guidelines, in order to make them more intuitive and easier to follow. During the preparation of the guidelines some rules had to be reformulated to make them more explicit and additional rules were added wherever necessary to better cover the practical annotation scenario and for being more complete.

The manual annotation task basically consisted of labeling or marking up manually through a customized web-interface the mentions of CHEMPROT interactions in text. Figure 1 summarizes the ChemProt relation types included in the annotation guidelines and ChemProt corpus.

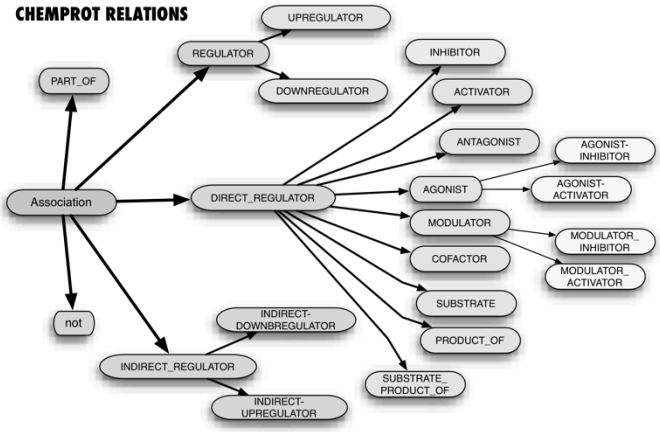


Fig. 1. Overview of the ChemProt track relation types and classification.

The annotation carried out for the ChemProt track was exhaustive for the types of interactions previously specified. This implied that mentions of other relationships between chemicals and genes (e.g. phenotypic and biological responses) were not annotated. The ChemProt relations are directed in the sense that only relations of "what a chemical does to a gene/protein" (chemical → gene/protein direction) were annotated, and not vice versa.

In order to establish a homogenous nomenclature and avoid redundant class definitions, we reviewed several chemical repositories that incorporate chemical – biology information. We thus inspected DrugBank (13), the Therapeutic Targets Database (TTD) (14) and ChEMBL (15), assay normalization ontologies (BAO) (16) and previously existing formalizations for the annotation of relationships: the Biological Expression Language (BEL) (17,18), curation guidelines for transcription regulation interactions (DNA-binding transcription factor – target gene interaction) and SIGNOR, a database of causal relationships between biological entities (19).

Each of these resources inspired the definition of the subclasses DIRECT REGULATOR (e.g. DrugBank, ChEMBL, BAO and SIGNOR) and the INDIRECT REGULATOR (e.g. BEL, curation guidelines for transcription

regulation interactions and SIGNOR). For example, DrugBank relationships for drugs included a total of 22 definitions, some of them overlapping with CHEMPROT subclasses (e.g. "Inhibitor", "Antagonist", "Agonist",…), some of them being regarded as highly specific for the purpose of this task (e.g. "intercalation", "cross-linking/alkylation") or referring to biological roles (e.g. "Antibody", "Incorporation into and Destabilization") and others, partially overlapping between them (e.g. "Binder" and "Ligand"), that were merged into a single class. Concerning indirect regulatory aspects, the five classes of casual relationships between a subject and an object term defined by BEL ("decreases", "directlyDecreases", "increases", "directlyIncreases" and "causesNoChange") were highly inspiring. Subclasses definitions of pharmacological modes of action were defined according to the UPHAR/BPS Guide to Pharmacology in 2016 (20,21).

For the CHEMPROT track a very granular chemical-protein relation annotation was carried out, with the aim to cover most of the relations that are of importance from the point of view of biochemical and pharmacological/biomedical perspective. Nevertheless, to simplify the CHEMPROT track, and to focus mainly on a subset of key relevant relation types, all the annotated CHEMPROT relations (CPRs) were grouped into 10 semantically related classes that do share some underlying biological properties. Those groups were labeled as [CPR:1, CPR:2, … CPR:10] ; and are detailed in Figure 2. For evaluation purposes only five groups labeled with 'Y' were used, that is: CPR:3, CPR:4, CPR:5, CPR:6, CPR:9.

Group	*Eval.*	*CHEMPROT relations belonging to this group*
CPR:1	N	PART_OF
CPR:2	N	REGULATOR|DIRECT_REGULATOR|INDIRECT_REGULATOR
CPR:3	Y	UPREGULATOR|ACTIVATOR|INDIRECT_UPREGULATOR
CPR:4	Y	DOWNREGULATOR|INHIBITOR|INDIRECT_DOWNREGULATOR
CPR:5	Y	AGONIST|AGONIST-ACTIVATOR|AGONIST-INHIBITOR
CPR:6	Y	ANTAGONIST
CPR:7	N	MODULATOR|MODULATOR-ACTIVATOR|MODULATOR-INHIBITOR
CPR:8	N	COFACTOR
CPR:9	Y	SUBSTRATE|PRODUCT_OF|SUBSTRATE_PRODUCT_OF
CPR:10	N	NOT

Fig. 2. Overview of the ChemProt track relation groups (CPRs).

Four different randomly sampled data subsets (Table 1) were released for the ChemProt track:

- An initial *sample set* of 50 abstracts to illustrate the type of chemical-protein relations and annotations that were used for the track, together with annotation guidelines for labeling the chemicals, genes/proteins and chemical-protein interaction relations. The sample set also contains an illustrative prediction example in the chemprot track format.
- A training set of 1,020 abstracts annotated exhaustively with chemicals, genes/proteins and chemical-gene interactions.
- A development set of 612 abstracts annotated exhaustively with chemicals, genes/proteins and chemical-gene interactions.
- A test set of 800 abstracts annotated exhaustively with chemical, genes/proteins and blinded chemical-gene interactions. Also, 2,599 additional abstracts were included in order to avoid manual corrections of team submissions and assure that systems could process larger datasets. Teams had to return results for the entire set of 3,399 records.

TABLE I. OVERVIEW OF CHEMPROT DATA SETS

Dataset	**Annotations**			
	Chemicals	***Genes***	***All CPI * relations***	***Evaluated CPI***
Sample	683	606	339	239
Training	13,017	12,735	6,437	4,157
Development	8,004	7,563	3,558	2,416
Test	10,810	10,018	5,744	3,458

* Chemical-protein interaction (CPI)

In practice, chemical-protein relation annotations prepared for the ChemProt track consisted of simple tab-separated columns containing:

1. Article identifier (PMID)
2. Chemical-Protein relation (CPR) group*
3. Evaluation type (Y: group evaluated, N: group not evaluated – extra annotation).
4. CHEMPROT relation (CPR)
5. interactor argument 1 (chemical entity followed by the interactor term identifier)
6. Interactor argument 2 (gene/protein entity followed by the interactor term identifier)

Track participants had to return for the collection of test set document identifiers the detected pairs of entities (one corresponding to a chemical entity and another to a gene/protein) together with the corresponding CPR group of the predicted relation. Only relations between a chemical and a gene/protein were allowed. Relations between a chemical and another chemical or between a gene/protein and another gene/protein were not permitted. Moreover participants were allowed to return for a given entity pair multiple relation groups. A total of 5 runs were accepted per team.

In addition to the ChemProt track data sets, a special evaluation script was available at the track webpage (22). For evaluation purposes we considered the micro-averaged precision, recall and, in particular, balanced micro F1-score.

III. RESULTS

The Markyt evaluation platform was used to register and upload the team submissions (23). A total of 13 teams returned overall 45 submissions for the ChemProt track. A detailed description of the underlying strategy used by each of the

participating teams can be found in the systems description papers published in the BioCreative VI workshop proceedings. Table 2 shows a summary of the participating teams.

TABLE II. OVERVIEW OF CHEMPROT PARTICIPATING TEAMS

Team Id	Details		
	Team Leader	***Institution***	***Nr. runs***
374	Sérgio Matos	Universidade de Aveiro	5
379	Sijia Liu	Mayo Clinic	4
394	Neha Warikoo	Academia Sinica	5
397	Atakan Yüksel	Boğaziçi University	1
403	Peter Corbett	Royal Society of Chemistry	1
404	Ignacio Tripodi	University of Colorado	5
417	Farrokh Mehryary	University of Turku	5
421	Cong Sun	DaLian University of Technology	1
424	Sangrak Lim	Korea University	2
427	Wei Wang	National University of Defense Technology	5
430	Yifan Peng	NCBI, NLM, NIH	5
432	Pat Verga	UMass Amherst	4
433	Pei-Yau Lung	Florida State University	2

In addition to evaluate all the team predictions, we prepared two simple baseline predictions. These consisted of predictions of all CPR classes for all the co-occurrence of chemical entities and genes, either within the entire abstracts, or within individual sentences. These baseline predictions can be considered as a sort of upper boundaries in terms of recall and lower boundaries in terms of precision when using simple entity co-mention. Table 3 illustrates the results obtained for each of the evaluated runs of all participating teams. Taking into account the complexity of the chemical-protein relations examined for this task, the results obtained by participating teams are very promising. The best F-measure, across all CPI relations, was reached by team 430 (run 5) with a score of 64.10%. Overall, the results were better in terms of precision when compared to recall, in particular for the top performing teams. Team 430 could obtain particularly high precision values for all runs, reaching up to 74.37% for run 1. The best recall values were obtained by team 403 (run 1) with a score of 67.84% followed by run 5 of team 427 (66.63%).

TABLE III. LIST OF CHEMPROT RESULTS PER TEAM AND RUN

Team Id	Run	Precision	Recall	F-Score
Co-mention	Abstract	0.0050	1.0000	0.0099
Co-mention	Sentence	0.0437	0.9803	0.0837
TEAM_374	RUN_1	0.6419	0.2577	0.3677
TEAM_374	RUN_2	0.5156	0.4670	0.4901
TEAM_374	RUN_3	0.5919	0.2403	0.3418
TEAM_374	RUN_4	0.4024	0.4193	0.4107
TEAM_374	RUN_5	0.5738	0.4722	0.5181
TEAM_379	RUN_1	0.4773	0.4375	0.4565
TEAM_379	RUN_2	0.4849	0.4913	0.4881
TEAM_379	RUN_4	0.5072	0.4306	0.4657
TEAM_379	RUN_5	0.5301	0.4639	0.4948
TEAM_394	RUN_1	0.2446	0.3407	0.2847
TEAM_394	RUN_2	0.2563	0.3456	0.2943
TEAM_394	RUN_3	0.2932	0.3271	0.3092
TEAM_394	RUN_4	0.0729	0.0150	0.0249
TEAM_394	RUN_5	0.2587	0.3456	0.2959
TEAM_397	RUN_1	0.6057	0.1102	0.1864
TEAM_403	RUN_1	0.5610	**0.6784**	0.6141
TEAM_404	RUN_1	0.3460	0.3913	0.3673
TEAM_404	RUN_2	0.3387	0.4078	0.3700
TEAM_404	RUN_3	0.3305	0.1666	0.2215
TEAM_404	RUN_4	0.3307	0.3641	0.3466
TEAM_404	RUN_5	0.3058	0.3603	0.3309
TEAM_417	RUN_1	0.6373	0.4462	0.5249
TEAM_417	RUN_2	0.6337	0.4387	0.5185
TEAM_417	RUN_3	0.6608	0.5662	0.6099
TEAM_417	RUN_4	0.6105	0.6006	0.6055
TEAM_417	RUN_5	0.6088	0.5989	0.6038
TEAM_421	RUN_1	0.1618	0.3409	0.2195
TEAM_424	RUN_1	0.6760	0.5159	0.5852
TEAM_424	RUN_2	0.6704	0.5194	0.5853
TEAM_427	RUN_1	0.2496	0.6417	0.3594
TEAM_427	RUN_2	0.2535	0.6478	0.3643
TEAM_427	RUN_3	0.2634	0.6622	0.3769
TEAM_427	RUN_4	0.2674	0.6602	0.3806
TEAM_427	RUN_5	0.2696	0.6663	0.3839
TEAM_430	RUN_1	**0.7437**	0.5529	0.6343
TEAM_430	RUN_2	0.7283	0.5503	0.6269
TEAM_430	RUN_3	0.7426	0.5382	0.6241
TEAM_430	RUN_4	0.7311	0.5685	0.6397
TEAM_430	RUN_5	0.7266	0.5735	**0.6410**
TEAM_432	RUN_1	0.2211	0.2024	0.2114
TEAM_432	RUN_2	0.5491	0.2021	0.2955
TEAM_432	RUN_3	0.4073	0.4783	0.4400
TEAM_432	RUN_4	0.4718	0.4453	0.4582
TEAM_433	RUN_1	0.6276	0.4858	0.5477
TEAM_433	RUN_2	0.6352	0.5121	0.5671

Performance varied also depending on the particular class of chemical-protein relations. For the antagonist relation class the best team obtained an F-measure of 72.56% (precision of 80.75%, recall of 65.87%) while for inhibition/down-regulation it was of 71.48% (with a precision of 76.51% and a recall of 67.07%).

IV. Discussion and Conclusions

The ChemProt track could engage a considerable number of teams, opening up further research on this topic. Systems resulting from this track can result in valuable contributions to improve the curation of chemical and biological data and promote the extraction of various types of chemical-protein interactions demanded by drug-discovery, metabolic reaction, gene regulatory network and systems biology analysis pipelines.

The ChemProt corpus comprises a large collection of manually annotated mentions of chemical compounds and genes/proteins that can serve to improve and validate bio-entity recognition tools, while the granular interaction types annotated for this task can foster more sophisticated bio-entity relation extraction pipelines. Only five general CPI classes were tested for this track, implying that the remaining relation classes and the actual granular relation types remain largely unexplored. We plan to promote additional efforts focusing on these more granular relation types, and provide the annotation of gene mention normalizations, i.e. biological database identifiers for the ChemProt corpus together with inter-annotator agreement results. When examining the underlying methodologies tested by participating teams, it becomes clear that machine learning techniques, and especially the use of artificial neural network approaches represent highly competitive strategies for chemical-gene interaction extraction tasks.

Acknowledgment

We acknowledge the encomienda MINETAD-CNIO/OTG Sanidad Plan TL and OpenMinted (654021) H2020 project for funding. We thank the Foundation for Applied Medical Research (FIMA) of the University of Navarra for financial support.

References

1. Krallinger, M., Leitner, F., Vazquez, M., Valencia, A. (2014) Text Mining. *Comprehensive Biomedical Physics,* **6**, 51-66.

2. Wei, C. H., Peng, Y., Leaman, R., Davis, A. P., Mattingly, C. J., Li, J., et al. (2016). Assessing the state of the art in biomedical relation extraction: overview of the BioCreative V chemical-disease relation (CDR) task. *Database*, *2016*.

3. Krallinger, M., Rabal, O., Lourenço, A., Oyarzabal, J., and Valencia, A. (2017). Information Retrieval and Text Mining Technologies for Chemistry. *Chem. Rev.*, **117** (12), 7673–7761

4. Rindflesch, T. C., Tanabe, L., Weinstein, J. N., and Hunter, L. (2000). EDGAR: extraction of drugs, genes and relations from the biomedical literature. In *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, 517–528.

5. Günther, S., Kuhn, M., Dunkel, M., Campillos, M., Senger, C., Petsalaki, E., et. al. (2007). SuperTarget and Matador: resources for exploring drug-target relationships. *Nucleic acids research*, ***36***(suppl_1), D919-D922.

6. Rebholz-Schuhmann, D., Kirsch, H., Arregui, M., Gaudan, S., Riethoven, M., and Stoehr, P. (2007). EBIMed—text crunching to gather facts for proteins from Medline. *Bioinformatics*, ***23***(2), e237-e244.

7. Tari, L., Hakenberg, J., Gonzalez, G., and Baral, C. (2009). Querying parse tree database of Medline text to synthesize user-specific biomolecular networks. In *Pacific symposium on biocomputing*, **14**, 87-98.

8. Czarnecki, J., Nobeli, I., Smith, A. M., and Shepherd, A. J. (2012). A text-mining system for extracting metabolic reactions from full-text articles. *BMC bioinformatics*, ***13***(1), 172.

9. Cañada, A., Capella-Gutierrez, S., Rabal, O., Oyarzabal, J., Valencia, A., & Krallinger, M. (2017). LimTox: a web tool for applied text mining of adverse event and toxicity associations of compounds, drugs and genes. *Nucleic Acids Research*. **45**, Issue W1, W484–W489.

10. Krallinger, M., Rabal, O., Leitner, F., Vazquez, M., Salgado, D., Lu, Z., et al. (2015). The CHEMDNER corpus of chemicals and drugs and its annotation principles. *Journal of cheminformatics*, **7**(S1), S2.

11. Perez, M. P., Rabal, O., Rodriguez, G. P., Vazquez, M., Fdez-Riverola, F., Oyarzabal, J., ... & Krallinger, M. (2017). Evaluation of chemical and gene/protein entity recognition systems at BioCreative V. 5: the CEMP and GPRO patents tracks. *Proceedings of the BioCreative*, *5*, 3-11.

12. Krallinger, M., Rabal, O., Lourenço, A., Perez, M. P., Rodriguez, G. P., Vazquez, M., et al. (2015). Overview of the CHEMDNER patents task. In Proceedings of the fifth BioCreative challenge evaluation workshop (pp. 63-75).

13. Wishart, D. S., Knox, C., Guo, A. C., Shrivastava, S., Hassanali, M., Stothard, P., et. al. (2006). DrugBank: a comprehensive resource for in silico drug discovery and exploration. *Nucleic acids research*, ***34***(suppl_1), D668-D672.

14. Yang, H., Qin, C., Li, Y. H., Tao, L., Zhou, J., Yu, C. Y., ... & Chen, Y. Z. (2015). Therapeutic target database update 2016: enriched resource for bench to clinical drug target and targeted pathway information. *Nucleic Acids Research*, **44**(D1), D1069-D1074.

15. Bento, A. P., Gaulton, A., Hersey, A., Bellis, L. J., Chambers, J., Davies, M., et al. (2014). The ChEMBL bioactivity database: an update. *Nucleic Acids Research*, **42**(D1), D1083-D1090.

16. Visser, U., Abeyruwan, S., Vempati, U., Smith, R. P., Lemmon, V., and Schürer, S. C. (2011). BioAssay Ontology (BAO): a semantic description of bioassays and high-throughput screening results. *BMC bioinformatics*, **12**(1), 257.

17. BEL Relationships. https://wiki.openbel.org/display/BLD/BEL+Relationships (accessed Mar 30, 2017).

18. BEL Causal Relationships. https://wiki.openbel.org/display/BLD/Causal+Relationships (accessed Mar 30, 2017)

19. Perfetto, L., Briganti, L., Calderone, A., Cerquone Perpetuini, A., Iannuccelli, M., Langone, F., ... & Peluso, D. (2016). SIGNOR: a database of causal relationships between biological entities. Nucleic acids research, 44(D1), D548-D554.

20. Southan, C., Sharman, J. L., Benson, H. E., Faccenda, E., Pawson, A. J., Alexander, S. P., et al. (2015). The IUPHAR/BPS Guide to PHARMACOLOGY in 2016: towards curated quantitative interactions between 1300 protein targets and 6000 ligands. *Nucleic Acids Research*, **44**(D1), D1054-D1068.

21. IUPHAR/BPS Guide to Pharmacology. http://www.guidetopharmacology.org/helpPage.jsp#ligand (accessed Mar 30, 2017).

22. http://www.biocreative.org/tasks/biocreative-vi/track-5/

23. Pérez-Pérez, M., Pérez-Rodríguez, G., Rabal, O., Vazquez, M., Oyarzabal, J., Fdez-Riverola, F., et al. (2016). The Markyt visualisation, prediction and benchmark platform for chemical and gene entity recognition at BioCreative/CHEMDNER challenge. *Database*, *2016*.

Chemical-protein relation extraction with ensembles of SVM, CNN, and RNN models

Yifan Peng[1], Anthony Rios[1,2], Ramakanth Kavuluru[2,3], Zhiyong Lu[1]
[1]National Center for Biotechnology Information, National Library of Medicine, National Institutes of Health, Bethesda, MD,
[2]Department of Computer Science, University of Kentucky, Lexington, KY
[3]Division of Biomedical Informatics, Department of Internal Medicine, University of Kentucky, Lexington, KY

***Abstract*—Text mining the relations between chemicals and proteins is an increasingly important task. The CHEMPROT track at BioCreative VI aims to promote the development and evaluation of systems that can automatically detect the chemical-protein relations in running text (PubMed abstracts). This manuscript describes our submission, which is an ensemble of three systems, including a Support Vector Machine, a Convolutional Neural Network, and a Recurrent Neural Network. Their output is combined using a decision based on majority voting or stacking. Our CHEMPROT system obtained 0.7266 in precision and 0.5735 in recall for an f-score of 0.6410, demonstrating the effectiveness of machine learning-based approaches for automatic relation extraction from biomedical literature. Our submission achieved the highest performance in the task during the 2017 challenge.**

Keywords—relation extraction; deep learning; chemical; protein

I. Introduction

Recognizing the relations between chemicals and proteins is crucial in various tasks such as precision medicine, drug discovery, and basic biomedical research. Biomedical researchers have studied various associations between chemicals and proteins and published their findings in biomedical literature. While manually extracting chemical-protein relations from biomedical literature is possible, it is often costly and time-consuming. Alternatively, text mining methods could automatically detect these relations effectively. The BioCreative VI track 5 CHEMPROT task [1] aims to promote the development and evaluation of systems that are able to automatically detect in running text (PubMed abstracts) relations between chemical compounds/drug and genes/proteins. In this paper, we describe our approaches and results for this task.

II. Methods

In the CHEMPROT track, the organizers developed a chemical-protein relation corpus composed of 4,966 PubMed abstracts, which were divided into a training set (1,020 abstracts), development set (612 abstracts) and test set (8,00 abstracts).

Unlike other relation corpora (1, 2), cross-sentence relations are rare in the corpus, appearing only in less than 1% in the training set. We also noticed that some chemical-protein pairs have multiple labels, but they only appear 10 times in the training set. As a result, our system treated the relation extraction task as a multiclass classification problem, and to simplify the problem, our system only focuses on the chemical-protein relations occurring in a single sentence.

[1] http://www.biocreative.org/tasks/biocreative-vi/track-5/

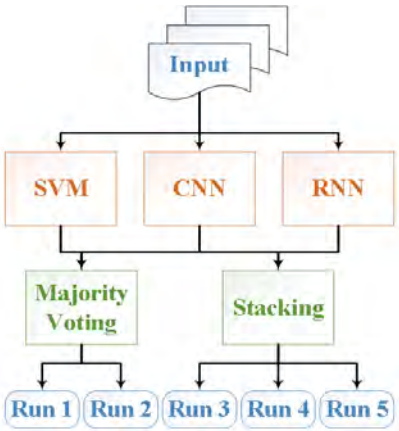


Fig. 1 Architecture of the systems for the CHEMPROT task

We addressed the CHEMPROT task using two ensemble systems that combine the results from 3 individual models, similar to our previous BioCreative submissions (3). An overview of the system architecture is shown in Figure 1. The individual systems included are a Support Vector Machine (SVM), a Convolutional Neural Network (CNN), and a Recurrent Neural Network (RNN) (4-6). We will describe these models together with the ensemble algorithms in the following subsections.

A. Rich Feature SVM

In our SVM system, the following features are exploited.

Words surrounding the chemical and gene mentions of interest: These features include the lemma form of a word, its part-of-speech tag, and chunk types. We used the Genia Tagger to extract the features (7). The window size is 5.

Bag-of-words between the chemical and gene mentions of interest in a sentence: These features include the lemma form of a word and its relative position to the target pair of entities (before, middle, after).

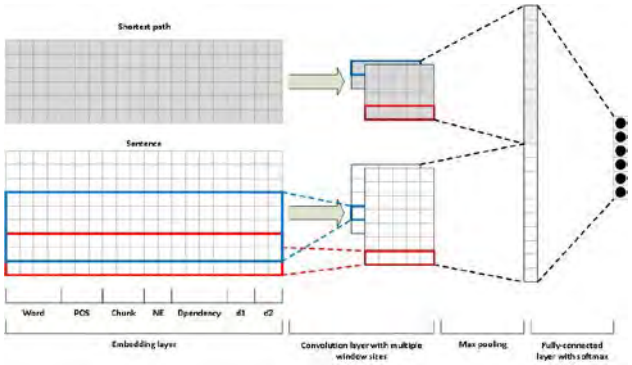


Fig. 2 Overview of the CNN model

The distance (the number of words) between two entity mentions in a sentence.

The existence of a keyword between two mentions often implies a specific type of a relation, such as "inhibit" for relation "CPR:4" and "agonism" for relation "CPR:5". Therefore, we manually built the keyword list from the training set and used them as features as well.

Shortest-path features include vertex walks (v-walks) and edge walks (e-walks) on the target pair in a dependency parse graph (8). An e-walk includes one word and its two dependencies. A v-walk includes two words and their dependency. For example, the shortest path of <Gemfibrozil, nitric-oxide synthase> extracted from the sentence "Gemfibrozil$_{CHEMICAL}$, a lipid-lowering drug, inhibits the induction of nitric-oxide synthase$_{GENE-N}$ in human astrocytes." is "Gemfibrozil ← nsubj ← inhibits → dobj → induction → nmod:of → nitric-oxide synthase". Thus the e-walks are "nsubj – inhibits – dobj" and "dobj – induction – nmod:of – nitric-oxide synthase". The v-walks are "Gemfibrozil – nsubj – inhibits", "inhibits – dobj – induction", and "induction – nmod:of – nitrix-oxide synthase".

We trained the SVM system using a linear kernel[2]. In our submissions, the penalty parameter C is 1 and the tolerance for stopping criteria is 1e-3. We also balanced the feature instances by adjusting their weights inversely proportional to class frequencies in the training set. We used one-vs-rest multiclass strategy.

B. Convolutional Neural Networks

We followed the work of Peng and Lu (9) to build the CNN model. Instead of using multichannels, we applied one channel but used two input layers (Fig. 2). One is the sentence sequence and the other is the shortest path between the pair of entity mentions in the target relation.

In our model, each word in either a sentence or a shortest path is represented by concatenating embeddings of its words, part-of-speech tags, chunks, named entities, dependencies, and positions relatively to two mentions of interest. The pre-trained word embedding vectors was learned on PubMed articles using the gensim word2vec implementation with the dimensionality set to 300 (10). The part-of-speech tags, chunks, and named entities were obtained from Genia Tagger (7).

[2] http://scikit-learn.org/

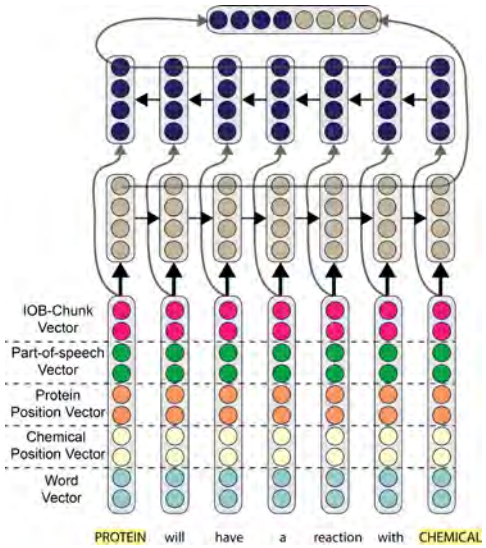


Fig. 3 Overview of the RNN model

The dependency information was obtained using the Bllip parser with the biomedical model and the Stanford dependencies converter (11-13). For each word, we used the dependency label of the "incoming" edge of that word in the dependency graph.

For each input layer, we applied convolution to inputs to get local features with window sizes of 3 and 5. ReLU function used as the activation unit and 1-max pooling was then performed to get the most useful global feature from the entire sentence.

In the fully connected layer, we concatenate the global features from both the sentence and the shortest path and then applied a fully connected layer to the feature vectors and a final softmax to classify the 6 classes (5 positive + 1 negative). We also used the dropout technique to prevent overfitting.

All parameters were trained using the Adam algorithm to optimize the cross entropy loss on a mini-batch with a batch size of 32 (14).

C. Recurrent Neural Netwoks

For our RNN model, we build on the work of Kavuluru et al. (15). Specifically, we train a bi-directional long-short term-memory (Bi-LSTM) recurrent model (Fig. 3), where the input to the model is a sentence. In this work, we don't consider the character level Bi-LSTM.

Similar to our CNN model, we concatenate the word embedding with the part-of-speech, IOB-chunk tag, and two position embeddings. The two position embeddings represent the relative location of the word with respect to the two entity mentions. It is important to note that we update the embeddings (word, part-of-speech, chunk, and position) during training.

After passing a sentence through our Bi-LSTM model, we obtain two hidden representations for each word. One representing the forward context, and the other representing the

backward. We concatenate the two representations to obtain the final representation of each word. To obtain a representation of the sentence, we use max-over-time (1-max) pooling across hidden state word representations.

Next, we pass the max-pooled sentence representation to a fully connected output layer. Unlike our CNN, we only apply a linear transformation without a softmax operation. Furthermore, the output layer only has 5 classes, where we completely discard the negative class. Specifically, we use the pairwise ranking loss proposed by Santos et al (16). Intuitively, the negative class will be noisy compared to the 5 positive classes. Rather than learning to predict the negative class explicitly, we force the 5 outputs to be negative. At prediction time, if all 5 class scores are negative, then we predict the negative class. Otherwise, we predict the class with the largest positive score.

Before training our model, we preprocess the dataset by replacing each word in the corpus that occurs less than 5 times with an unknown (UNK) token. Also, given each instance is comprised of a sentence and two entity mentions, we replace each entity with the tokens CHEMICAL or PROTEIN dependent on what the specific mention represents.

Finally, we train our RNN model using the Adam optimizer with a mini-batch size of 32. For the Adam optimizer, we set the learning rate to 0.001, beta1 to 0.9 and beta2 to 0.999. In addition, we apply recurrent dropout of 0.2 in the Bi-LSTM model and standard dropout of 0.2 between the max-pooling and output layers. We use pre-trained word vectors (6B Token GLOVE[3]) with a dimensionality of 300. Likewise, the POS, position, and chunk vectors are randomly initialized, and each has a dimensionality of 32. We should note that both the POS and chunk tags are extracted using NLTK[4]. Lastly, we set the hidden state size of the LSTM models to 2048.

D. A Majority Voting System

In the voting system, we combined the results of the three models using a majority voting. That is, we select the relations that are predicted by more than 2 models.

E. A Stacking System

While voting is a straightforward way to combine our SVM, CNN, and RNN models, methods that are more sophisticated can improve our performance. Specifically, we use stacking to combine the predictions of each model. Stacking works by training multiple base models (SVM, CNN, and RNN), then trains a meta-model using the base model predictions as features.

For our meta-model, we train a Random Forest (RF)[5] classifier. First, in order to train the RF, we capture the scores for each class from all 3 models on the development set. In total, we have the following 17 features: 6 from the SVM, 6 from CNN, and 5 from the RNN (because we used a ranking loss). For the CNN scores, we use the unnormalized scores for each class before passing them through the softmax function. Finally, we train the RF on the development set using 50,000 trees and the gini splitting criteria.

[3] https://nlp.stanford.edu/projects/glove/
[4] http://www.nltk.org/
[5] http://scikit-learn.org

III. Results and Discussion

In the CHEMPROT track, the test set contains 800 abstracts.

Our submissions were prepared with an ensemble of 3 models. We built every SVM, CNN and RNN model using 80% total data (training + development) and built the ensemble system using the remaining 20% of the total data. To reduce variability, 5-fold cross-validation was performed using different partitions of the data. As a result, we obtained 5 SVMs, 5 CNNs, and 5 RNNs in total. We submitted 5 runs as our final submissions. Runs 1 and 2 use a majority voting system and Runs 3-5 use a stacking system. Each run uses one SVM, CNN, and RNN from one cross validation iteration.

Table 1 shows the overall performance of the ensemble system as reported by the organizer, where 'P', 'R', 'F' denotes precision, recall, and F1 score, respectively.

TABLE I. Results for our ensemble system on test set

Run	System	P	R	F
1	Majority Voting	0.7437	0.5529	0.6343
2	Majority Voting	0.7283	0.5503	0.6269
3	Stacking	0.7426	0.5382	0.6241
4	Stacking	0.7311	0.5685	0.6397
5	Stacking	0.7266	0.5735	0.6410

IV. Conclusion

In this manuscript, we describe our submission in the BioCreative VI CHEMPROT task. The results demonstrate that our ensemble system can effectively detect the chemical-protein relations from biomedical literature. We also show that the domain specific features are useful for in this task.

Acknowledgment

The authors thank the organizers of the BioCreative VI CHEMPROT task. This research is supported by the Intramural Research Programs of the National Institutes of Health, National Library of Medicine. A.R. was a summer intern at the NCBI/NIH and supported by the NIH Intramural Research Training Award. R.K.'s involvment is supported by the National Library of Medicine through grant R21LM012274.

References

1. Li,J., Sun,Y., Johnson,R., Sciaky,D., Wei,C.H., Leaman,R., Davis,A.P., Mattingly,C.J., Wiegers,T.C., Lu,Z. (2015) Annotating chemicals, diseases and their interactions in biomedical literature. In *Proceedings of the BioCreative V Challenge Evaluation Workshop*.173-182.

2. Wei,C., Peng,Y., Leaman.R., Davis,A.P., Mattingly,C.J., Li,J., Wiegers,T.C., Lu,Z. (2016) Assessing the state of the art in biomedical relation extraction: overview of the BioCreative V chemical-disease relation (CDR) task. *Database (Oxford)*.1-8.
3. Leaman,R., Wei,C.H., Zou,C. Lu,Z. (2016) Mining chemical patents with an ensemble of open systems. *Database (Oxford)*, baw065.
4. Vapnik,V. (1995) The nature of statistical learning theory. Springer-Verlag New York, Inc..
5. LeCun,Y., Bottou,L., Bengio,Y., Haffner,P. (1998) Gradient-based learning applied to document recognition. *Proceedings of the IEEE*. **86**, 2278-2324.
6. Hochreiter,S., Schmidhuber,J. (1997) Long short-term memory. *Neural Computation*. **9**, 1735-1780.
7. Tsuruoka,Y., Tsujii,J. (2005) Bidirectional inference with the easiest-first strategy for tagging sequence data. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT-EMNLP)*. 467-474.
8. Kim,S., Yoon,J., Yang,J. (2007) Kernel approaches for genic interaction extraction. *Bioinformatics*. **24(1)**, 118-126.
9. Peng,Y., Lu,Z. (2017) Deep learning for extracting protein-protein interactions from biomedical literature. In *Proceedings of the 2017 Workshop on Biomedical Natural Language Processing (BioNLP)*. 29-38.
10. Mikolov,T., Chen,K., Corrado,G., Dean,J. (2013) Efficient Estimation of Word Representations in Vector Space. In *the International Conference on Learning Representations (ICLR)*. 1-12.
11. McClosky,D. (2009). Any domain parsing: automatic domain adaptation for natural language parsing. Department of Computer Science, Brown University.
12. Charniak,E. (2000) A maximum-entropy-inspired parser. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*. 132-139.
13. De Marneffe,M., Dozat,T., Silveira,N., Haverinen,K., Ginter,F., Nivre,J., Manning,C. (2014) Universal Stanford dependencies: A cross-linguistic typology. In *Proceedings of 9th International Conference on Language Resources and Evaluation (LREC)*.**14**, 4585-4592.
14. Kingma,D., Ba,J. (2015) Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*. 1-15.
15. Kavuluru,R., Rios,A., Tran,T. (2017) Extracting Drug-Drug Interactions with Word and Character-Level Recurrent Neural Networks. In *Proceedings of the 2017 International Conference on Health Informatics (ICHI)*. 5-12.
16. Santos,C., Xiang,B., Zhou,B. (2015) Classifying Relations by Ranking with Convolutional Neural Networks. In *proceedings of the 53rd Annual Meeting of the Association of Computational Linguistics (ACL).* 626–634.

Extracting Chemical-Protein Interactions using Long Short-Term Memory Networks

Sérgio Matos
Affiliation: DETI/IEETA, University of Aveiro, Aveiro, Portugal

***Abstract*—Automatic relation extraction from biomedical literature is required for collecting structured information from the growing number of published works. Interactions between chemicals and genes or proteins are particularly important, namely for expediting work on precision medicine and drug discovery. In this paper, we present an approach based on long short-term memory networks with syntactic information for extracting chemical-gene relations from scientific abstracts. The proposed approach achieved an f-score of 0,5181 on the BioCreative VI CHEMPROT task corpus.**

Keywords—relation extraction; chemical-gene interactions; deep learning; long short-term memory networks

I. Introduction

Relation extraction is a text mining task that aims at identifying textual mentions of relations between entities. In biomedical text mining, the entities of interest are diseases, proteins, chemicals and drugs, anatomical regions, symptoms, among others. Automatically extracting these relations from the large number of scientific publications available is of utmost importance for collecting the most recent research results and for expediting database curation.

Previous research on biomedical relation extraction focused mostly on protein-protein interactions (PPI), gene-disease, and chemical-disease relations (1, 2). For PPI extraction, kernel based methods have been shown to provide good results. These methods extract a representation of sentences and then apply a kernel function to compare sentence pairs using this representation. A comprehensive study on five different PPI corpora showed that kernel methods based on dependency parsing achieved area-under-the-curve (AUC) results ranging from 0.81 to 0.85 for within corpus cross-validation, and from 0.69 to 0.87 for cross-learning evaluation (3). The recent BioCreative V CDR task evaluated the extraction of chemical-disease relations. In this task, participants were also required to identify the disease and chemical entities, in addition to extracting the relations. The best method achieved an f-score of 0.57 using a combination of sentence and document level features and domain knowledge (4). Zhou et al. (5) and Gu et al. (6) employed deep learning classifiers based on long short-term memory (LSTM) and convolution neural network (CNN), with dependency features. Both works achieved an f-score of 0.61, although using gold-standard entities.

Deep learning techniques are composed of multiple data transformation layers that apply simple non-linear functions to obtain different levels of representation of the input data (7). Combination of these transformations allow the methods to learn complex classification functions, which gives deep learning its strengths. Another great advantage of such representation learning is that it eliminates the feature engineering effort that is required in more traditional machine learning (7). These methods have been applied successfully in many tasks, most notably in image classification but also in several NLP tasks, including word sense disambiguation (8), text classification (9), and named entity recognition (10).

For text based tasks, it is necessary to encode the input data in a way that it can be used by the deep network classifier. This is achieved by representing words as embedding vectors of a relatively small dimension, rather than using the large feature space resulting from the traditional one-hot encoding. Word embeddings is a technique that consists in deriving vector representations of words, such that words with similar semantics are represented by vectors that are close to one another in the vector space (11). This way, each document is represented by a sequence of word vectors which are fed directly to the network. Efficient calculation of word embeddings, such as provided by Word2Vec (12), allow inferring word representations from large unannotated corpora.

The CHEMPROT task, part of the BioCreative VI community challenge, was to evaluate systems for automatically extracting relations between chemicals and gene in PubMed abstracts. This paper describes our participation in this task, based on the application of LSTM recurrent neural networks with features extracted from the dependency structure of the sentences.

II. Methods

A. Data

The CHEMPROT corpus is composed of a training set with 1020 abstracts, a development set with 612 abstracts, and a test set with 800 abstracts. These documents were manually labeled by experts who identified all chemical and gene mentions, and all chemical-gene relations according to a set of biologically relevant relation classes. For the task, five different classes were considered. Table I shows the statistics of the dataset.

B. Preprocessing

We preprocessing each dataset using TEES (13) (https://github.com/jbjorne/TEES) to obtain the dependency structure of all sentences. This process can take around 3.5 to 4 seconds per document, when executed on a single processing core.

This work was supported by FCT - Foundation for Science and Technology, in the context of the project IF/01694/2013.

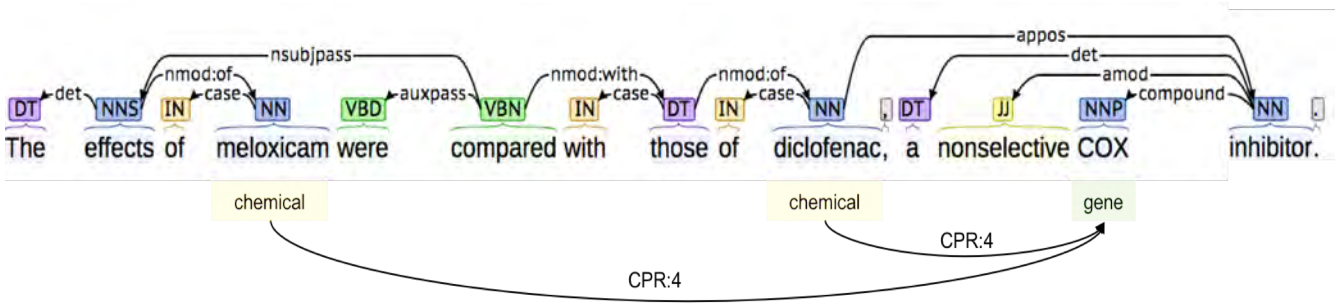


Fig. 1. Example sentence illustrating the dependency structure and chemical-gene relations.

TABLE I. DATASET STATISTICS

Entities	Training	Development
Chemical	13017	8004
Gene	12735	7563
Relations	**Training**	**Development**
upregulation/activation	768	550
downregulation/inhibition	2254	1094
agonist	173	116
antagonist	235	199
substrate	727	457

Then, for each entity pair in each sentence, we obtain the shortest dependency path connecting the entity mentions and extracted the followings six sequences:

1) words in the path;
2) POS tags of words in the path;
3) path dependencies;
4) up to 30 words before the first entity;
5) the words between the entities;
6) up to 30 words after the second entity.

Figure 1 illustrates the dependency structure of a sentence from the dataset. In this example, there are three entities (*meloxicam*, *diclofenac* and *COX*) and two relations. Our preprocessing stage extracts two instances from this sentence, one for each relation, and each instance is represented by the six sequences mentioned above. For example, for the relation between *meloxicam* and *COX*, the shortest path leads to the following sequences, where the participating entities are replaced by the words 'chemical' and 'gene':

Words in path: chemical effects compared those diclofenac inhibitor gene

POS in path: NN NNS VBN DT NN NN NN

Dependencies in path: prep_of nsubjpass prep_with prep_of appos nn

We extracted a training instance for each possible chemical/gene pair in each sentence. Pairs that did not correspond to a gold standard relation were used as negative training samples. For entities spanning more than one word, we obtained the shortest path starting from the head word. The process for the test data was similar.

These sequences are fed to the LSTMs through embedding layers. For the words in the shortest path and words before/between/after the entities, we used word embeddings pre-calculated from the full MEDLINE. These embedding vectors are kept fixed during training. POS and dependency embeddings were randomly initialized and adapted during trained.

C. Word Embeddings

We used the word2vec implementation in the Gensim framework (14) and generated word embeddings from the complete MEDLINE database, corresponding to 15 million abstracts in English language. We created six models, with vector sizes of 100 and 300 features and windows of 5, 20 and 50. The models contain around 775 thousand distinct words. Based on previous results both in document triage and biomedical word sense disambiguation (15), we used the model with 300 features and window size of 50.

D. Deep Learning

Convolutional neural networks (CNN) have been extensively applied in image recognition and classification problems, with very good performance. Various works also demonstrate their application in text classification tasks. Nonetheless, the sequential nature of natural texts can be better modelled by recurrent networks, which contain a feedback loop that allows the network to use information regarding the previous state. LSTM are a special type of recurrent neural networks (RNN) in which a set of information gates is introduced that allow these networks to learn long-term dependencies while avoiding the vanishing gradient problem.

We used a network composed of 3 to 6 bidirectional LSTM networks, followed by a fully connected layer and by an output layer with softmax activation (Figure 2). The inputs to each LSTM are obtained from the shortest dependency path between the chemical and gene entities in a sentence, using the provided gold standard entities.

An important consideration when defining and training deep network models is related to overfitting, which means that the

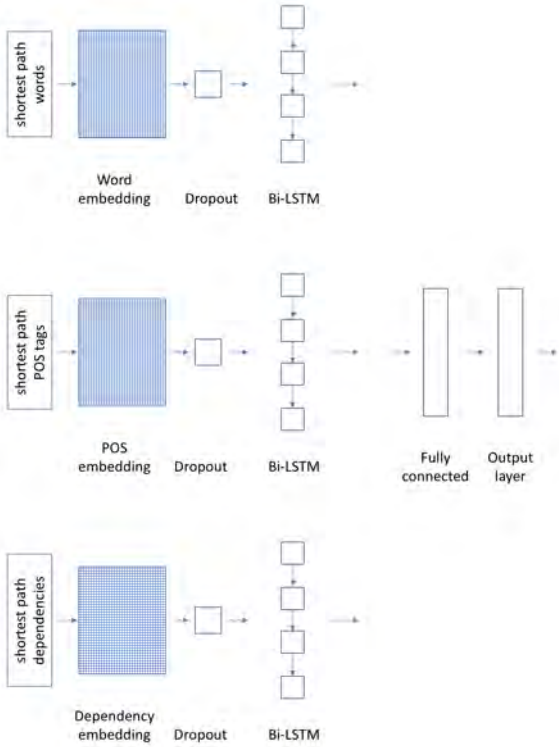


Fig. 2. Neural network structure.

network learns the "best" data representation but is not able to generalize to unseen data. Various strategies have been proposed and are commonly employed to address this problem. In our experiments, we applied several strategies to avoid overfitting, namely early stopping, dropout, and regularization. Early stopping looks at the value of the loss in a validation subset and stops the training process when this value stops decreasing. We used 10% of the training data, selected randomly at each run, as validation set. We applied dropout with rate 0.2 to the embedding and 0.1 to the LSTM layers so that a random selection of the output tensors is not used for updating the model weights, with the aim of forcing the model to learn a less biased representation of the data. Finally, L2 regularization with parameter 0.01 was applied to the final layer to penalize large weights that could otherwise be assigned to biased input dimensions.

E. Submitted Runs

Table II shows the configuration of the submitted runs. All configurations use the sequence of words, POS tags and dependencies in the shorted path connecting the two entities. Configuration 1 and 2 use also the sequence of words between the two entities, and configuration 5 uses this and the words before the first entity and after the second entity in the sentence. Furthermore, given that the classes are highly unbalanced, configurations 2, 4 and 5 were trained with class weights that are the reciprocal of their frequency relative to the majority class, that is, the negative class. All models were trained using the binary cross-entropy loss function and the Adam algorithm as optimizer. Models were implemented with the Keras framework (16) with the TensorFlow backend and executed on a machine with 12 CPU cores and 192 GB of memory.

TABLE II. CONFIGURATION OF SUBMITTED RUNS

Run	Configuration						
	Dependency features			Sentence features			Class weights
	Word	*POS*	*Dep*	*Left*	*Middle*	*Right*	
1	x	x	x		x		
2	x	x	x		x		x
3	x	x	x				
4	x	x	x				x
5	x	x	x	x	x	x	x

TABLE III. RESULTS ON DEVELOPMENT SET AND OFFICIAL RESULTS ON TEST SET

Run	Development			Test		
	Precision	*Recall*	*F-Score*	*Precision*	*Recall*	*F-Score*
1	0,6547	0,5403	0,5919	**0,6419**	0,2577	0,3677
2	0,4856	0,6221	0,5449	0,5156	0,4670	0,4901
3	0,6334	0,5126	0,5664	0,5919	0,2403	0,3418
4	0,4310	0,6092	0,5047	0,4024	0,4193	0,4107
5	0,4999	0,6074	0,5470	0,5738	**0,4722**	**0,5181**

III. RESULTS

Table III shows the results on the development set and official evaluation results on the test set. The development set results shown are the average of 5 training runs. As expected, adding class weights improves recall since less frequent classes are weighted up. Also, adding word sequences before, between and after the entities improved the results.

Comparing the results on the development set against those on the test set, there is a drop in precision of up to 4 percentage points, expect for run 5, while recall drops considerably by 18 to 28 points. Interestingly, run 5 originated an increase of 7 percentage points in precision in the test set compared to the development set. Error analysis could give some indication on how generalization could be improved.

REFERENCES

1. Krallinger,M., Vazquez,M., Leitner,F., et al. (2011) The Protein-Protein Interaction tasks of BioCreative III: classification/ranking of articles and linking bio-ontology concepts to full text. BMC bioinformatics, 12(8), S3.

2. Wei,C.H., Peng,Y., Leaman,R., et al. (2016) Assessing the state of the art in biomedical relation extraction: overview of the BioCreative V chemical-disease relation (CDR) task. Database, 2016.

3. Tikk,D., Thomas,P., Palaga,P., Hakenberg,J. and Leser,U. (2010) A comprehensive benchmark of kernel methods to extract protein–protein interactions from literature. PLoS computational biology, 6(7), e1000837.

4. Xu,J., Wu,Y., Zhang,Y., Wang,J., Lee,H.J. and Xu,H. (2016) CD-REST: a system for extracting chemical-induced disease relation in literature. Database, 2016.

5. Zhou,H., Deng,H., Chen,L., Yang,Y., Jia,C. and Huang,D. (2016) Exploiting syntactic and semantics information for chemical–disease relation extraction. Database, 2016.

6. Gu,J., Sun,F., Qian,L. and Zhou,G. (2017) Chemical-induced disease relation extraction via convolutional neural network. Database, 2017.

7. LeCun,Y., Bengio,Y. and Hinton,G. (2015) Deep learning. Nature, 521(7553), 436-444.

8. Yepes, A.J. (2017) Word embeddings and recurrent neural networks based on Long-Short Term Memory nodes in supervised biomedical word sense disambiguation. Journal of Biomedical Informatics, 73, 137-147.

9. Kim,Y. (2014) Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882.

10. Habibi,M., Weber,L., Neves,M., Wiegandt, D.L. and Leser,U. (2017) Deep learning with word embeddings improves biomedical named entity recognition. Bioinformatics, 33(14), i37-i48.

11. Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. (2003). A neural probabilistic language model. Journal of machine learning research, 3(Feb), 1137-1155.

12. Mikolov,T., Chen,K., Corrado,G. and Dean,J. (2013) Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

13. Björne,J. and Salakoski,T. (2015) TEES 2.2: biomedical event extraction for diverse corpora. BMC bioinformatics, 16(16), S4.

14. Rehurek,R. and Sojka,P. (2010). Software framework for topic modelling with large corpora. In In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks.

15. Antunes,R. and Matos,S. (accepted). Supervised learning and knowledge-based approaches applied to biomedical word sense disambiguation. Journal of Integrative Bioinformatics.

16. Chollet,F. et al. (2015) Keras. https://github.com/fchollet/keras

Attention-based Neural Networks for Chemical Protein Relation Extraction

Sijia Liu[1,2], Feichen Shen[1], Yanshan Wang[1], Majid Rastegar-Mojarad[1,3], Ravikumar Komandur Elayavilli[1], Vipin Chaundary[2], Hongfang Liu[1]

[1]*Department of Health Sciences Research, Mayo Clinic, Rochester, MN, USA*
[2]*Department of Computer Science and Engineering, University at Buffalo, Buffalo, NY, USA*
[3]*University of Wisconsin-Milwaukee, Milwaukee, WI, USA*

***Abstract*— Relation extraction is an important task in the field of natural language processing and text mining. In this paper, we described our participation in Biocreative VI Task 5: Text mining chemical-protein interactions (CHEMPROT). We used deep neural networks, including convolutional neural networks and attention based recurrent neural networks, to extract chemical protein relationships. We submitted four runs as final submission, with the best performing micro average F1 score of 0.506 on the development set and 0.494 on the test set. Our experimental results indicate that attention based RNN models outperform the CNN models. The replacement of raw texts into semantic types (chemical, gene) also improved the performance.**

Keywords—relation extraction, natural language processing, recurrent neural networks, attention mechanism

I. Introduction

Efficient and accurate access to the information from scientific literature is among the most widely studied topics in text mining. To facilitate and promote information extraction systems on PubMed abstracts, Biocreative VI CHEMPROT task aims to automatically detect the relations among chemical compounds, genes and proteins. There have been several chemical- or drug-related relation extraction share tasks organized in recent years. Two DDI-Extraction share tasks were organized to extract the drug-drug interaction information from knowledge base (DrugBank) and scientific articles (Medline) in 2011 and 2013, respectively [1]. In 2015, Biocreative V Track 3 targeted the extraction of relationships in Chemical-Induced Diseases (CID) [2]. Chemical Protein interaction (Chemprot) is an interesting topic for biomedical relation extraction [3].

Deep Neural Network (DNN) models have been used for relation extraction tasks recently, including Convolutional Neural Networks (CNN) [4] and Recurrent Neural Networks (RNN) [5]. Moreover, attention mechanism on top of deep neural networks has shown promising results in various Natural Language Processing (NLP) tasks, such as machine translation [6], question answering [7] and document classification [8]. Attention based model has been used in multiple relation extraction tasks [9][10]. In this paper, we described our participation in Biocreative VI Task 5: Text mining chemical-protein interactions (CHEMPROT). We used deep neural networks, including CNN and attention based RNN to extract chemical protein relationships. We achieved a micro average F1 score of 0.506 on the development set and 0.494 on the test set as our best result. The experimental results indicated that attention based Gated Recurrent Units (GRU) outperforms other DNN models.

II. Materials

We used the ChemProt corpus provided by task organizers. The corpus consisted of 4966 PubMed abstracts with 126,457 annotated chemical and protein entities. The relations were annotated with 10 chemical-protein relations. According to the share task description, only 5 out of 10 semantic relation types would be evaluated. Therefore, we focused only on the relation groups included in the official evaluation (CPR 3, 4, 5, 6 and 9).

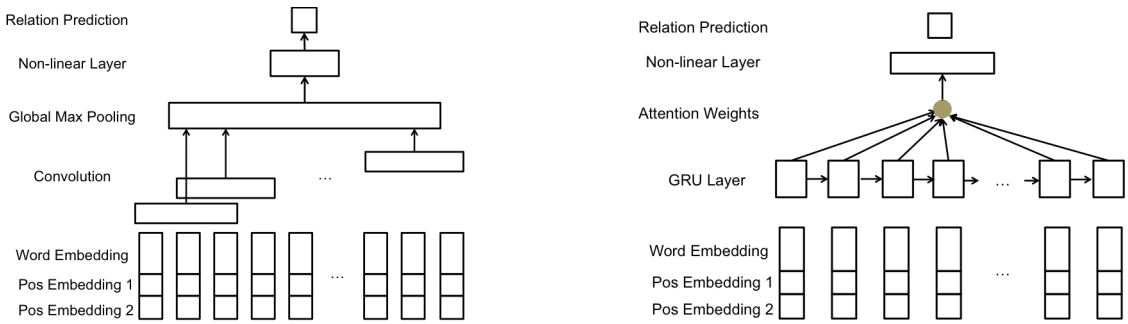
Table I shows the corpus statistics of the training, development and testing datasets, including the number of documents in the dataset, the average number of entities per document (abstract) and the average number of positive relations per document. The test set gold standard relation annotation was not available by the time we prepared the manuscript. More details of the datasets can be found in the CHEMPROT task description paper.

TABLE I CHEMPROT CORPUS STATISTICS

Dataset	# of docs	Average # of entities	# of positive relations	# of all potential relations
Training	1020	25.247	4157	15842
Development	612	25.436	2416	9759
Test	3334	25.536	-	-

III. Method

We used DNN for the task of relation extraction on the annotated named entity mentions. We model the relation extraction problem as a relation classification problem among all the chemical-protein relation pairs within one sentence. The rationale of limiting the relations into within-sentence relations is that we only found three and six cross-sentence relations from training and development sets, respectively. Therefore, on



(a) (b)

Figure 2. Model architecture of CNN

the CHEMPROT corpus, eliminating all cross-sentence relations will not impact the performance significantly.

A. Relation Instance Generation

We extract the sentences from sentence detectors. For each potential chemical protein pair in the sentence, we assign a relation label "NA" for the pair without annotated gold standard annotation provided by the challenge organizers. Here we consider the relations other than the five evaluated types as negative relations (CPR 1, 2, 7, 8 and10).

B. Input Representation

In our proposed system, the input to the neural network models is expected to be low dimension semantic token-level vectors. We use word embeddings and position embeddings as the input to the neural network models.

We use two different methods to represent the pre-annotated entities. In the first method, the tokens are directly sent to the word embedding model to retrieve the word embeddings, regardless if it is an annotated token as part of an entity. The limitation for this method is that a lot of chemical, gene and protein entities may not be found from the pre-trained model. We then apply a second representation. We replace all the entity tokens by the entity types. Specifically, for chemical entity mentions, all the tokens of entities are replaced by the word token "chemical". Similarly, all the gene and protein entities are replaced by "gene", regardless of whether it can be normalized or associated to a biological database identifier (i.e. "GENE-Y" or "GENE-N").

Word embeddings. We use 300-dimension pre-trained Glove-6B model. Our preliminary experiments show that the 300 dimensional Glove-6B outperforms the word embedding models we trained by CBOW from PubMed. If a word cannot be found from the word embedding model, the embedding will be generated randomly and appended into the model.

Position embeddings. We follow the method by Zeng et. al. [4] to generate the position embedding of the entities in each narrative sentence. The position embedding is generated based on the relative distances of tokens to the entities. An example of relative distance is shown in Figure 2.

For each relation instance, there will be two position embeddings for each token from two entities (one chemical and one gene/protein). The two position embeddings are

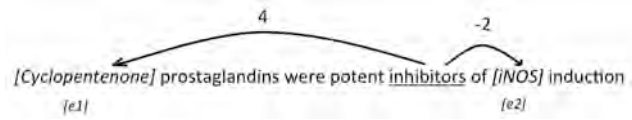


Figure 1. Example of relative distance of the word "inhibitors"

concatenated to the word embeddings of the token as the input to the neural network models.

C. Convolutional neural network

Our CNN models for relation extraction are similar to [4]. The model architecture is shown in Figure 1 (a). The convolutional layer can capture contextual information of filters of a pre-defined filter length. The convolutional filters are expected to generate high-level local features from the input vector representations. The output of the convolutional layer is then forwarded to the Global Max-pooling layer, where the maximum values of each filter outputs are pooled and concatenated. Finally, a non-linear function can be used on the max-pooling vector to predict the probability-like values of each relation label. We then select the label with the highest value from the non-linear layer as the relation label.

D. Attention based RNN

Attention mechanism is proposed to emphasize the contribution of the informative neural units in the model. Instead of directly receiving the signals from consecutive RNN units, the additional attention layer overlooks all the RNN units of the sequence and assigns different attention weights to each unit according to their importance. The intuition for applying attention-based model is to have higher weights for tokens that are indicators of the semantic relations. We use the equations from Luong, et. al. [6] to calculate attention weights for each token in a sentence. The output of the RNN units is denoted as $h = [h_1, h_2, \dots, h_T]$, where T is the sentence length. Given token representation as w, we define the hidden representation of h_t as u_t and word level importance vector u_w. The relation representation vector s is the weighted sum of RNN outputs h and the attention weights α.

$$u_t = \tanh(W_w h_t + b_w)$$

$$\alpha_t = \frac{\exp(u_t^T u_w)}{\sum_t \exp(u_t^T u_w)}$$

$$s = \sum_t \alpha_t h_t$$

where W_w and b_w are the weight matrix and bias of the attention layer, respectively. We then apply the non-linear layer and relation predication method as described in the CNN model (Section III.C) to predict relation labels.

E. Evaluation

The model performance is evaluated via standard micro-average F1-score. True Positive (TP) denotes the number of correctly detected positive relation instances. False Positive (FP) denotes the number of relations "NA" in the gold standard but are predicted as one of the positive relations by the DNN model. False Negative (FN) denotes the number of positive instances that are not detected by the model. The micro-average F1-score is defined as:

$$Precision = \frac{TP}{TP + FP},$$

$$Recall = \frac{TP}{TP + FN},$$

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}.$$

The system performance is evaluated by the official evaluation package provided by the task organizers[1].

IV. Result

A. Experimental settings

We used Punkt sentence detector in NLTK [11], [12] to detect sentence boundaries. The models are implemented using Keras 2.0.5[2] with Tensorflow[3] backend. The implementation of attention layer is inspired by [13]. We applied the hyperparameters shown in Table II for the CNN and ATTention-based GRU model (ATT-GRU). The models are trained using Adam optimizer on the loss function of sparse categorical cross entropy. Dropout was applied to the non-linear layers to prevent overfitting [14]. The dropout rate was set to 0.5.

TABLE II PARAMETER SETTINGS FOR CNN AND ATT-GRU MODELS

Parameter	Value
Batch size	64
Number of filters	100
Filter length	3
Hidden dimensions	200
RNN units	128
Learning rate	0.001

B. Submitted Runs

We submitted four runs to the challenge evaluation. The details of each run is shown as follows:

Run 1: CNN with raw tokens as input, without replacing annotated entity tokens.

Run 2: ATT-GRU raw tokens as input, without replacing annotated entity tokens.

Run 3: CNN with raw tokens replaced as labels.

Run 4: ATT-GRU with raw tokens replaced by entity labels.

We also implemented conventional RNN, GRU, LSTM, and attention based LSTM as preliminary experiments. However, the performance was not as good as the performance of ATT GRU models. Due to the limitation of the number of submission runs, we did not include them into the official submissions.

Table IV shows the system performance of each submitted run in the development dataset. Based on our experimental results, the attention based GRU on entity labels outperforms the rest of runs with a F1 score of 0.506. Attention based GRU systems outperform CNN models. Replacing raw entity texts by the entity labels enhances the models slightly for both CNN and ATT-GRU. It is mainly due to the fact that most of the chemical, gene and protein tokens cannot be found in the pre-trained word embedding models. Thus, replacing them into meaningful entity labels ("chemical", "gene") can help find reasonable word embeddings instead of random assigned ones. The models generally do not suffer much from overfitting. All the submission runs observed F1 score decreases of less than 0.01.

The performance breakdown of our best run (ATT-GRU) on development set is shown in Table IV. The classification report is done via scikit-learn[4]. CPR:4 has the highest performance among all the classes, and has the largest proportion in all relations as well. CPR:3 is one of the hardest relation type to classify, not only for the ATT-GRU model but also for our other models.

TABLE III PERFORMANCE BREAKDOWN OF ATT-GRU ON THE DEVELOPMENT SET

Label	Support	Precision	Recall	F1-score
CPR:3	498	0.473	0.388	0.426
CPR:4	990	0.569	0.663	0.613
CPR:5	112	0.357	0.634	0.457
CPR:6	184	0.505	0.609	0.552
CPR:9	407	0.468	0.442	0.455
Total *	2191	0.512	0.553	0.528

* The "Total" line is calculated from weighted F1-score by support counts, which is different from the micro-F1 score of the official evaluation.

V. Conclusion and Future Work

In this paper, we described our participation of Task 5: Text mining chemical-protein interactions (CHEMPROT). We developed deep neural network models including CNN and attention-based RNN using the embedding of either raw entity mention tokens or annotated entity type labels as input. The model using annotated entity type labels and attention-based GRU model achieves the best performance on both the

[1] http://www.biocreative.org/media/store/files/2017/evaluation-kit.zip

[2] https://keras.io/

[3] https://github.com/tensorflow/tensorflow

[4] http://scikit-learn.org/stable

TABLE IV SYSTEM PERFORMANCE OF SUBMITTED RUNS ON DEVELOPMENT SET

Run ID	Model	Development Set			Test Set		
		Precision	Recall	F1-score	Precision	Recall	F1-score
1	CNN token	0.459	0.456	0.457	0.477	0.437	0.456
3	CNN entity	0.497	0.448	0.471	0.507	0.430	0.465
	ATT-LSTM token	0.429	0.485	0.456	-	-	-
2	ATT GRU token	0.470	**0.522**	0.494	0.484	**0.491**	0.488
4	ATT GRU entity	**0.512**	0.501	**0.506**	**0.530**	0.463	**0.494**

development and the test set, with F1-scores of 0.506 and 0.494, respectively.

There are some directions to investigate to improve this work. We would like to see if an external knowledge base can be used to improve our machine learning based system, which is dependent on the provided corpus. Another interesting problem is how to utilize the token-level weights from the attention activations. The weights can be further applied for tasks such as pattern mining and key word extraction. The current system is proposed for within-sentence relations. The model could be further extended into abstract level by introducing hierarchical neural network models [8].

ACKNOWLEDGMENT

The authors would like to thank the support of the U.S. National Institutes of Health grant R01 LM011829 and an NSF IPA grant. Opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Institutes of Health and National Science Foundation.

REFERENCE

[1] I. Segura-Bedmar, P. Martínez, and M. Herrero-Zazo, "Lessons learnt from the DDIExtraction-2013 Shared Task," *J. Biomed. Inform.*, vol. 51, pp. 152–164, 2014.

[2] C. H. Wei *et al.*, "Assessing the state of the art in biomedical relation extraction: Overview of the BioCreative V chemical-disease relation (CDR) task," *Database*, vol. 2016, 2016.

[3] M. Krallinger, O. Rabal, A. Lourenço, J. Oyarzabal, and A. Valencia, "Information retrieval and text mining technologies for chemistry," *Chemical Reviews*, vol. 117, no. 12. pp. 7673–7761, 2017.

[4] D. Zeng, K. Liu, S. Lai, G. Zhou, and J. Zhao, "Relation Classification via Convolutional Deep Neural Network," *Coling*, no. 2011, pp. 2335–2344, 2014.

[5] A. Katiyar and C. Cardie, "Investigating LSTMs for Joint Extraction of Opinion Entities and Relations," *Proc. 54th Annu. Meet. Assoc. Comput. Linguist. (ACL 2016)*, pp. 919–929, 2016.

[6] M.-T. Luong, H. Pham, and C. D. Manning, "Effective Approaches to Attention-based Neural Machine Translation," Aug. 2015.

[7] Y. Xiang, Q. Chen, X. Wang, and Y. Qin, "Answer Selection in Community Question Answering via Attentive Neural Networks," *IEEE Signal Process. Lett.*, vol. 24, no. 4, pp. 505–509, 2017.

[8] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical Attention Networks for Document Classification," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 1480–1489.

[9] G. Ji, K. Liu, S. He, and J. Zhao, "Distant Supervision for Relation Extraction with Sentence-Level Attention and Entity Descriptions," *Ai*, pp. 3060–3066, 2017.

[10] P. Zhou *et al.*, "Attention-Based Bidirectional Long Short-Term Memory Networks for Relation Classification," *Proc. 54th Annu. Meet. Assoc. Comput. Linguist. (Volume 2 Short Pap.*, pp. 207–212, 2016.

[11] T. Kiss and J. Strunk, "Unsupervised Multilingual Sentence Boundary Detection," *Comput. Linguist.*, vol. 32, no. 4, pp. 485–525, 2006.

[12] S. Bird and E. Loper, "NLTK: The Natural Language Toolkit," in *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, 2004, pp. 1–4.

[13] R. Liao, "Text Classification, Part 3 - Hierarchical attention network," 2016. [Online]. Available: https://richliao.github.io/supervised/classification/2016/12/26/textclassifier-HATN/.

[14] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *J. Mach. Learn. Res.*, vol. 15, pp. 1929–1958, 2014.

Extracting chemical-protein interactions from literature

Pei-Yau Lung[1], Tingting Zhao[2], Zhe He[3], Jinfeng Zhang[1]
[1]Department of Statistics, Florida State University, Tallahassee, FL, 32306 USA
[2]Department of Geography, Florida State University, Tallahassee, FL, 32306 USA
[3]College of Communication and Information, Florida State University, Tallahassee, FL, 32306 USA

***Abstract* — Chemical compound and protein interaction information is indispensable for understanding the regulation of biological processes and development of therapeutic drugs. Manually extracting such information from biomedical literature is very time and resource consuming. In this study, we proposed a computational method to automatically extract chemical-protein interactions (CPI) from a given text. Our method extracts CPI-pairs and CPI-triplets from sentences, where a CPI-pair is defined as a chemical compound and a protein name in a sentence, and a CPI-triplet is defined as a chemical compound name, a protein name, and a word describing their relationship (interaction word) in a sentence. We formulate the task as a machine learning problem and extract a diverse set of features from sentences, which were used to build multiple machine learning models. Our models contain both simple features, which can be directly computed from sentences and also more sophisticated features derived using natural language processing (NLP) techniques. For example, one set of features are extracted based on the shortest paths between the CPI-pairs or among the CPI-triplets in the dependency graphs obtained from sentence parsing. To predict the multiple categories of CPI types, we designed a three-stage approach to divide-and-conquer the challenging multi-class prediction problem. Our method obtained good performance on the BioCreative VI track 5 data with F1-scores of 0.7641, 0.7912, and 0.7798 for train set, development set and the combination of them, respectively, on 10-fold cross validation.**

Keywords—Protein-protein interactions; chemical-protein relations; text mining; CPI triplets;

I. INTRODUCTION

Chemical-protein interactions (CPI) play important roles in metabolism and regulation of biological processes. Such knowledge is also the key to understanding mechanisms of diseases and developing therapeutic drugs. With many years of biological and biomedical research, a large amount of information on CPI has been discovered and published in scientific literature as unstructured text. Text information stored in the literature is not convenient for automatic analysis and computation and a great deal of efforts has been made to manually extract such information from literature and deposit it in various databases as structured data. However, manual extractions of chemical-protein interaction information is very time and resource consuming and it can be difficult to keep up with the speed of scientific publications. Consequently, computational methods that can automatically extract the desired information have been sought to address the problem.

A considerable number of computational methods have been developed to extract protein-protein interaction information (PPI) from literature in the past (1-11). These approaches tackled the problem from varies aspects, ranging from relatively simple co-occurrence, to rule-based pattern matching, to machine learning based methods, which can be further enhanced by natural language processing (NLP) techniques. However, far less methods have been proposed to automatically extract the interactions or relations between genes/proteins and chemical compounds/drugs. A good chemical-protein interaction extraction method will benefit not only basic biomedical research but also precision medicine as well as drug discovery. Such methods can be used to extract unstructured CPI information from literature and deposit it to databases to allow efficient query of CPI information. The structured CPI information can also be used in integrative analysis of other biomedical datasets. A CPI extraction method can also be used to extract CPI from a given text to get information on very specific types of chemical-protein interactions.

To promote research in this important biological text mining area, BioCreative challenge VI has established a separate track for this task, BioCreative VI track 5, Text mining chemical-protein interactions. As a participant of this challenge, we have developed a method for extracting CPI information from biomedical literature. This manuscript describes our method and findings in this study. In this track, the goal is to build a system to automatically detect or extract the relations between chemical compounds/drugs and genes/proteins in PubMed abstracts. CPI extraction is challenging because we have to find whether chemical compound and protein mentions have interaction, as well as to identify which type of interaction they share. To develop the CPI extraction method, we model CPI extraction as a classification problem. By employing varies sets of features from semantic pattern and structure of the sentence with several machine learning classifiers, our method is capable to identify the CPI within biomedical literature. The results using datasets released by BioCreative VI demonstrate that our method is effective in CPI extraction.

The rest of the paper is organized as following. We first describe the proposed method. The data used in this study and the experimental results are then presented, which is followed by conclusion and discussions.

II. METHODS

Our CPI extraction method extracts potentially true CPI-pairs and CPI-triplets from a given text. The CPI-pair is defined as a chemical compound name and a protein name in the same sentence and CPI-triplet is defined as a chemical compound name, a protein name and an interaction word in the same sentence. Our CPI extraction method builds models on CPI-pairs and CPI-triplets, then combines them to form a final model. We first construct all possible CPI-pairs and CPI-triplets from entities mentions in a sentence. For each pair or triplet, we next extract feature sets from both the semantic pattern and the dependency-graph of the sentence. Semantic pattern contains the information of how people describe the interaction between chemicals and proteins, and dependency graph provides the information on how words are interconnected in the sentence. These features are used in machine learning methods to build models, which will predict the corresponding relation group of a CPI-pair or a CPI-triplet.

A. Chemical-protein pairs and triplets construction

The BioCreative VI track 5 has tagged all chemical compounds and proteins/genes mentioned in an abstract. We used these tagged names to construct CPI-pairs and CPI-triplets. A CPI-pair will be labeled as one of ten CPI groups (CPR:[1-10]) according to the relation annotations provided by BioCreative VI. Each group has its own underlying biological properties. If a CPI-pair is constructed but not included in the relation annotations, it will be labeled as group 10 (CPR:10).

A CPI-triplet is defined as the combination of a CPI-pair with one interaction word, which is used to describe the relation of interaction between the chemical compound and the protein. We have manually built an interaction words dictionary based on a previous study (10), which contains 1,107 interaction words. These interaction words were further manually labeled to the corresponding CPI group. If a CPI-pair is in a sentence where there are multiple interaction words, the same number of CPI-triplets will be constructed, and these CPI-triplets share the same label of such CPI-pair.

B. Features for model building

1) Semantic pattern

We extracted features based on our experience and through reading some sentences in the training and development data sets. For example, a CPI-pair is likely to be a certain CPI group if there are such type of interaction words included in the region covered by the pair. Another example is that when a CPI-pair is CPR:9, some related words, e.g. pathway, production, generate, synthesis, are frequently mentioned in three words before and after the chemical compound or protein/gene mentions.

We also employed features proposed by (10). These features capture the information of certain grammar or language rules that people used to describe the PPI. We found they are helpful in CPI extraction as well. For example, a binary feature, that whether negative words such as *not*, *incapable* and *unable* are in the region covered by the pair or triplet, captures the negative meaning. Likewise, a binary feature, that whether sentence breaking words such as *although*, *therefore*, *whereas*, etc., are in the region, captures the compound meanings in the sentence.

2) Dependency graph

The Stanford Dependency Parser was used to analyze the structure of the sentence. The parser also provides the dependency graph of a sentence. One direct way to extract the relation of two words is to find the shortest dependency path (SDP) in the graph (12). It is reasonable to assume that the SDP contains the necessary and dense information, helpful for identifying the relation of chemical compound and protein/gene mentioned in a sentence. Fig. 1. shows an example of a dependency graph for the sentence "*Binding studies showed that the first TPR motif of SGT interacts with the UbE motif of the GHR.*" The SDP between the two protein names, SGT and GHR, are colored as yellow. The texts colored as red, such as *nn*, *nsubj*, are typed dependencies.

To avoid the unnecessary complexity of dependency graph for better parsing accuracy, the names of chemical compound and protein/gene in a CPI-pair were replaced with CHEM and PROT respectively when parsing a sentence. Other entities mentioned in the sentence were replaced with special symbols "CPT + *interactor term identifier*". Table I shows an example of sentence after replacing the entities names.

For CPI-pair, we obtained the SDP between CHEM and PROT. For CPI-triplet, two additional SDP, CHEM to interaction word and interaction word to PROT, were obtained. We next extracted features from the SDP. For example, the distance (number of words in between) of SDP of CHEM to PROT. This distance tends to have small value when there is relation between CHEM and PROT. In addition, type dependencies in SDP are also helpful for distinguishing the true CPI from false ones. For example, when SDP of CHEM to PROT has small value of distance as well as *appos* (appositive) dependency, CHEM and PROT tend to be apposition, indicating they have no interaction. Therefore, we added binary features for type dependencies included in the SPD.

TABLE I. ENTITIES NAMES REPLACEMENT

Sentence	PMID	Arg1	Arg2
	14507899	**T15**	**T16**
Before	**P2Y(2) receptor** agonist **INS37217** enhances functional recovery after detachment caused by sub**retinal** injection in normal and rds mice.		
After	**PROT** agonist **CHEM** enhances functional recovery after detachment caused by sub**CPT10** injection in normal and rds mice.		

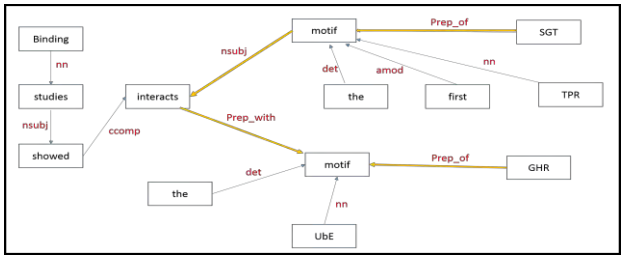


Fig. 1. Grammatical dependencies graph

C. Three-stage model building

We implemented a three-stage model building approach. At each stage, several models were built on CPI-pairs and CPI-triplets respectively. Outcomes of those models will be used as additional feature sets at the next stage.

1) Stage I

At stage I, we built models for binary classification to predict whether a CPI-pair or CPI-triplet is true CPI, which is defined as CPR:[3, 4, 5, 6, 9]. These five groups carry different meaning for the CPI, and such information can be useful for downstream applications. For example, it is important to know whether an interaction is activation or inhibition type. Three classifiers were used in this stage: Random Forest, Extremely Randomized Trees, and Gradient Boosted Trees model. The predictions and rank-normalized predicted probabilities were stored and used at stage II.

2) Stage II

Combining the original feature set with the outcomes from stage I, we built models for multiclass classification to predict which CPI group a CPI-pair or CPI-triplet belong to. Three additional classifiers, Logistic Regression, Linear Discriminant Analysis and Naive Bayes, were used. The predicted probabilities for each CPI group, and the CPI group with highest predicted probability, from each classifier were stored. For those CPI-triplets from a CPI-pair due to multiple interaction words, we chose the one which gives the highest predicted probability as the outcome for such CPI-pair. Table II shows an example of choosing a CPI-triplet constructed from the CPI-pair in Table I.

TABLE II. EXAMPLE FOR CHOOSING CPI-TRIPLET

Sentence	PROT agonist CHEM enhances functional recovery after detachment caused by subCPT10 injection in normal and rds mice.		
CPI-Triplet	***Prediction***	***Predicted Probability***	***Choose?***
CHEM-PROT-agonist	CPR:5	0.7234	Yes
CHEM-PROT-enhances	CPR:3	0.5118	No
CHEM-PROT-caused	CPR:9	0.3841	No

3) Stage III

At the final stage, three feature sets were combined: original features on CPI-pair, and outcomes from CPI-pairs and CPI-triplets at stage II. Gradient Boosted Trees model, as well as feature selection and optimization for tuning parameters, were implemented to get the final models used for predicting the cases in the test data set.

III. RESULTS

A. Data

The data used in this study are from PubMed articles, and divided into train set and development set by BioCreative VI track 5 organizers. There are totally 1,632 abstracts, in which 41,319 entities are tagged. We extracted 18,229 CPI-pairs and 51,460 CPI-triplets from the train set, and 11,397 CPI-pairs and 32,150 CPI-triplets from development set. Table III reports the number of pairs and triplets in each true CPR group.

TABLE III. NUMBER OF CASES IN EACH TRUE CPR GROUP

Train set	***CPR:3***	***CPR:4***	***CPR:5***	***CPR:6***	***CPR:9***
# pairs	761	2,251	173	235	727
# triplets	2,492	6,452	486	732	1,536
Development set	**CPR:3**	**CPR:4**	**CPR:5**	**CPR:6**	**CPR:9**
# pairs	548	1,093	115	199	457
# triplets	1,650	2,936	400	673	881

B. Results

To test our models, we used train set as training data, then made predictions on the development set as testing data. We also did the other way around. The performance was evaluated by F_1-score, taking micro-average across true CPR. Table IV reports the F_1-score of the models at Stage III. F_1-scores of training data were computed using 10-fold cross validation. Our model can reach up to 0.7912 F_1-score in the 10-fold cross validation for the training data. The F_1-score in testing data was 0.5796 for development set, and 0.5397 for train set (when development set was used as training data). It is notable that our three-stage model building approach boosts the performance. The best performance on testing data at Stage II is 0.5702 for development set, and 0.5291 for train set. The considerable discrepancy in F_1-score between training data and testing data indicates that the train set and the development set might be quite different in certain ways.

TABLE IV. F_1-SCORE OF MODELS AT STAGE III

Training data	***10-fold CV F_1-score***	***Testing data***	***F_1-score***
Train set	0.7641	Development set	0.5796
Development set	0.7912	Train set	0.5397
Train set + Development set	0.7798	-	-

IV. CONCLUSION

In this study, we proposed a method to detect chemical compounds-proteins interactions (CPI) mentioned in biomedical literature. We constructed CPI-pairs and CPI-triplets, extracted varies sets of features from semantic pattern and structure of the sentence, and implemented a three-stage approach with several machine learning methods. When tested on BioCreative VI track 5 data, our method achieved satisfactory performance, indicating that it can be used in practical CPI extraction tasks.

There are several directions we can take to further improve our method. First, we can extract features containing more and deeper semantic information shared by various dataset. This can be facilitated by reading some cases for which we have failed to make the right classification. Second, we may manually annotate more cases and add them to the training data. Third, when reading some of the mistakes we have made for development data, we found that a significant number of cases were wrongly labeled. Correcting wrongly labeled cases may further improve our method. Fourth, we have labeled all the CPI-triplets formed by the same CPI-pair and different interaction word in a sentence as the same chemical-protein interaction type, which certainly has some cases wrongly labeled. This can be addressed in the future by selecting one and only one of them to assign the right label since this is probably right for most of the cases. This may be achieved by predicting the one to be labeled and conducting the training iteratively. Finally, we can try some deep learning based method as deep learning has been shown to be quite effective in recent years in many different types of machine learning problems including text mining.

ACKNOWLEDGMENT

We thank Albert Steppi for helpful discussions. This research was supported by the National Institute of General Medical Sciences of the National Institute of Health under award number R01GM126558.

REFERENCES

1. Airola A*, et al.* (2008) All-paths graph kernel for protein-protein interaction extraction with evaluation of cross-corpus learning. *BMC bioinformatics* 9(11):S2.
2. Chang Y-C, Chu C-H, Su Y-C, Chen CC, & Hsu W-L (2016) PIPE: a protein–protein interaction passage extraction module for BioCreative challenge. *Database* 2016.
3. Giuliano C, Lavelli A, & Romano L (2006) Exploiting shallow linguistic information for relation extraction from biomedical literature. *EACL*, (Trento Italy), pp 401-408.
4. Miwa M, Sætre R, Miyao Y, & Tsujii Ji (2009) Protein–protein interaction extraction by leveraging multiple kernels and parsers. *International journal of medical informatics* 78(12):e39-e46.
5. Qian L & Zhou G (2012) Tree kernel-based protein–protein interaction extraction from biomedical literature. *Journal of biomedical informatics* 45(3):535-543.
6. Sætre R, Sagae K, & Tsujii Ji (2007) Syntactic features for protein-protein interaction extraction. *LBM (Short Papers)* 319.
7. Van Landeghem S, Saeys Y, De Baets B, & Van de Peer Y (2008) Extracting protein-protein interactions from text using rich feature vectors and feature selection. *3rd International symposium on Semantic Mining in Biomedicine (SMBM 2008)*, (Turku Centre for Computer Sciences (TUCS)), pp 77-84.
8. Yang Z*, et al.* (2011) Multiple kernel learning in protein–protein interaction extraction from biomedical literature. *Artificial intelligence in medicine* 51(3):163-173.
9. Bui Q-C, Katrenko S, & Sloot PM (2010) A hybrid approach to extract protein–protein interactions. *Bioinformatics* 27(2):259-265.
10. Chowdhary R, Zhang J, & Liu JS (2009) Bayesian inference of protein–protein interactions from biological literature. *Bioinformatics* 25(12):1536-1542.
11. Niu Y, Otasek D, & Jurisica I (2009) Evaluation of linguistic features useful in extraction of interactions from PubMed; application to annotating known, high-throughput and predicted interactions in I2D. *Bioinformatics* 26(1):111-119.
12. Bunescu RC & Mooney RJ (2005) A shortest path dependency kernel for relation extraction. *Proceedings of the conference on human language technology and empirical methods in natural language processing*, (Association for Computational Linguistics), pp 724-731.

Knowledge-base-enriched relation extraction

Biocreative VI Task 5: Text mining chemical-protein interactions

Ignacio Tripodi[1], Mayla Boguslav[2], Negacy Hailu[2], Lawrence E. Hunter[2]

1: University of Colorado, Boulder, BioFrontiers Institute. 2: University of Colorado Denver, Computational Bioscience Program

Prior knowledge about how a chemical interacts with genes or proteins might be valuable in predictive computational toxicology. Many relationships between chemicals and proteins (or genes) have been catalogued in various databases. However, these databases are incomplete; some information can be found only in the literature. Here, we describe a feature engineering and relation classification approach that leverages information in databases to improve the quality of relation extraction with the goal of identifying relationships missing from those databases.

Automated relation extraction from text is difficult due to the many ambiguities in natural language. The current state of the art consists of extracting features such as words, word stems, and syntactic information, and using them as inputs to a machine learning classifier. Here, we explore whether automatic identification of relationships between chemicals and proteins found in publications can be enriched by adding prior knowledge about the chemicals and proteins found in existing databases to the features used in machine learning.

Keywords: Relation extraction, knowledge-bases, natural language processing, biomedical ontologies.

I. Introduction

Our approach was to combine features from the text with information from a knowledge-base. We integrate knowledge from many different databases using the KaBOB (1) knowledge-base, to automatically identify a set of five possible relations ("upregulation", "downregulation", "antagonist", "agonist", and "substrate of") between a chemical and a protein mentioned in PubMed abstracts. The knowledge-base incorporates information about the chemicals and proteins (i.e. "participates in kinase activity", "has N aromatic rings", "it's lipoxygenase activating", etc). We tested our approach on an extensive manually annotated set of relations from the ChemProt (2) database (including therapeutics), using this prior knowledge in conjunction with text-derived features.

II. Materials

We used the following tools during preprocessing and feature extraction:

1. Headword finder: Many of the chemical and protein/gene names are multiwords. For such names, we used Michael Collins' Headword Finder implementation in the Stanford CoreNLP to find the heads. Headword words are important in extracting dependency path features.
2. Dependency parser: we trained SyntaxNet on the CRAFT (3) corpus. The model was tested on unseen CRAFT set and it achieved state-of-the-art results. We used the dependency parser to extract two features --- to find dependency label path between two pair of entities, and to find words along the dependency path
3. TFIDF Vectorizer: Text features are converted into number using scikit-learn's TFIDF Vectorizer implementation.

III. Methods

One aspect of this task was to determine whether there was a relation between a chemical and protein/gene entity in the first place. We made the assumption that any mention of an annotated chemical and protein in the same sentence represented a potential relation. Of course, this is not the case for every possible chemical-protein pair, so we had to train a machine learning classifier to detect when there was no relation. To achieve this, we pooled the sample, training and development data as our "training dataset", and for every sentence in every abstract, if we saw a chemical and protein annotated within it, we created a training sample. If there was indeed a relation defined for these two, we set the training label as such; otherwise, we set it as "NONE". Any relation that wasn't part of the list of interest for this task was also labeled as "OTHER". Random 80/20 splits were performed to evaluate performance and help tune the classifier settings.

For each of these CHEMICAL/GENE-[Y|N] pairs, we gathered the following features from the text itself, following in part suggestions from Jurafsky et al. (4) Relation Extraction chapter:

- The tokenized chemical entity name, and its bigrams
- The tokenized protein entity name, and its bigrams
- The tokenized combined chemical and protein entity names, and bigrams
- The tokenized words in between mentions, and bigrams
- The number of words between mentions
- The tokenized words of the sentence this relation happens in, and bigrams
- The tokenized head word for the first occurring entity, and bigrams
- The tokenized head word for the second occurring entity, and bigrams
- Dependency parser labels between the chemical and protein, including bigrams and trigrams
- The words along the dependency path from the first occurring entity to the second occurring entity, including bigrams and trigrams

In addition, when it was possible to normalize the chemicals and/or proteins to a ChEBI ID and Protein Ontology ID (respectively), features were extracted from KaBOB for both the chemical and protein,. The feature set from KaBOB was constructed as follows:

- A vector representing all possible GO annotations associated with a protein, and all possible ChEBI classes formed the basis of the feature set. Very abstract and very rare features were removed from this vector
- If a protein could be normalized to KaBOB, the vector positions for that protein's GO annotations were set to 1. If a chemical could be normalized to KaBOB, then the vector positions for its ChEBI entry and all its IS-A parents were set to 1. Other positions were set to 0. If the protein or chemical did not map to an entity in KaBOB, the respective positions were set to 0.5.
- During error analysis we found a subset of the KaBOB features of interest, that seemed highly relevant during successful relation classification. When those features were present, they were set to 2, (rather than 1).

We executed classification of the chemical-protein relations using a variety of machine learning algorithms:

- Naive Bayes
- Perceptron (100 iterations)
- Random Forests
    - A grid search was performed to obtain the most favorable settings for 100 estimators.
    - After performing 500 estimators and noticing no significant improvement, it was determined that 100 estimators were enough
    - Perceptron and Neural Networks ultimately displaced this algorithm in performance.
- Neural Networks
- Feature selection was performed using a chi-squared test of f_classif (ANOVA) method. After utilizing a selection of the best 10k, 20k and 30k features, it was observed that the full dataset without feature selection performed better, therefore this approach was abandoned. We also intended to use SURF (an extension of ReliefF) to extract the best features, but it proved not to scale in time due to the vast number of features.

IV. Results

Below are our results as evaluated by the organizers:

TABLE I. Results on the test set from the organizers

Run	Performance Metrics		
	Precision	***Recall***	***F-Score***
Team 404 Run 1	0.3460	0.3913	0.3673
Team 404 Run 2	0.3387	0.4078	0.3700
Team 404 Run 3	0.3305	0.1666	0.2215
Team 404 Run 4	0.3307	0.3641	0.3466
Team 404 Run 5	0.3058	0.3603	0.3309

These results in Table I are low compared to what we were achieving on a 20% of the training set. Table-II shows our best model results evaluated against 20% of the training set. This could be due to various reasons. Firstly, the test set is sizeable compared to the training set. Since most of our features are text-based, a larger test set could introduce lots of unseen words in the training set that will result in lower results in the test set. Secondly, we might have overfit our models.

TABLE II. Results from our best model on 80/20 split

Relation	Performance Metrics		
	Precision	***Recall***	***F-Score***
CPR:3	0.76	0.74	0.75
CPR:4	0.79	0.80	0.80
CPR:5	0.60	0.60	0.60

CPR:6	0.61	0.74	0.67
CPR:9	0.75	0.83	0.79
NONE	0.92	0.91	0.91
OTHER	0.72	0.75	0.73
AVG / TOTAL	0.86	0.86	0.86

The best performance was achieved using neural networks with one hidden layer. The input and hidden layer have 200 nodes. We found that more layers and nodes were achieving better results but the training time was too long. Since we had limited time, we decided for one hidden layer network. The outer layer has seven nodes since there were seven classes in the training set. Overfitting is a big problem when training neural networks. To address this issue, we used dropout with a value of 0.5. Our experiments were conducted using Keras framework.

We used different metrics to evaluate the performance of our development runs: for each type of score (precision, recall and F1-score) we calculated the weighted average, the micro-average, and macro-average of each category.
The models we selected to submit after testing a wide array of combinations, all exclude the tokens derived from the full sentence in which the chemical and protein entities are present and instead use dependency parsing. Further, all models use the dependency parser labels as features. The main differences between the models include the machine learning algorithm, bigrams or trigrams from words in the dependency pattern, and use of knowledge-base-derived features. The models are:

1. Neural network on all text features using unigrams, bigrams and trigrams for the words along dependency path between the two entities.
2. Neural network on all text features using unigrams and bigrams for the words along dependency path between the two entities.
3. Neural network on all text and knowledge-base-derived features using unigrams and bigrams for the words along dependency path between the two entities.
4. Perceptron neural network using all text features using unigrams, bigrams and trigrams for the syntax dependency parser output words.
5. Naive Bayes neural network using all text features using unigrams, bigrams and trigrams for the syntax dependency parser output words.

A. Incorporating the knowledge-base

A significant portion of finding related attributes of chemicals and proteins in KaBOB was mapping the annotated strings to an identifier represented in the knowledge-base. For the purposes of this task, we attempted to match chemical strings to a ChEBI ID, and proteins/genes to a Protein Ontology ID or gene ontology molecular function (GO MF) ID.. We employed various heuristics to find these mappings, always using the case-insensitive string denoting the entity to match against:

- ChEBI chemical names
- ChEBI synonym for chemicals
- CAS numbers in ChEBI's accession data
- PubChem chemical names
- KEGG compound names

For all the above we attempted to match the entity string verbatim, replacing Greek letters by their romanized name, replacing numbers with Roman numerals, adding "compound" as a suffix, splitting multiple terms with spaces, and splitting multiple terms with a dash. The protein lookups were performed against the Protein Ontology attempting to match with a protein name or a synonym first and then the gene ontology molecular function family as a last resort. Various heuristics were employed for protein searches: matching the string verbatim, adding " (human)" as a suffix, adding a " protein" suffix, adding a "-like protein" suffix, using spaces only as separators, using "-" only as separators, adding a "h" prefix, removing all punctuation, and removing the "human" identifier portion of the name. As a last resort, if there were no matches to the Protein Ontology, "activity" was added to the protein name as a suffix and searched for in the molecular function gene ontology, which provides a large set of gene/gene product functionalities (e.g., kinase activity), and many genes/gene products are either named or can be referred to as entities that possess these functionalities (e.g., kinase). Plus, in using the gene ontology molecular function, we can take advantage of the extensive hierarchy for reasoning and machine learning.

V. Error Analysis

Once we had some significant success in classifying the chemical and protein interactions, the false positives and false negatives were examined for possible improvements to either the text or knowledge-based-derived features. For the text features, misclassifications of the opposite relations, including a chemical-protein pair classifying as both upregulating and downregulating, were troublesome and appeared to be due to

both relationships being in the text, but for different chemical-protein pairs. For example, "DBDCT up-regulated the expression of Bax, down-regulated the expression of Bcl-2, and significantly increased the ratio of Bax/Bcl-2." Both "up-regulated" and "down-regulated" are in the same sentence but only apply to specific proteins, Bax and Bcl-2 respectively. This led to using the dependency path instead of the full sentence for the text features. For false negatives, the main issues were with the "OTHER" category and due to it being a collection of all the non-relevant categories, teasing out the issue was unclear. Thus we were not able to figure out why recall is low in order to improve it, but we did improve precision and F1-score by using the dependency path.

For the knowledge-based-derived features, we determined the unique chemical and protein attributes specific to each of the five possible relations, if any. All relations had at least 1 unique feature except for "agonist" which had no unique protein features and only one unique chemical feature. This information was aggregated into a list of features of interest that were weighted 2 in the feature matrix. Overall, both of these updates, the dependency path instead of the full sentence and the list of KaBOB features of interest, were used in the final algorithms submitted to improve performance.

VI. Discussion

The words in the entire sentence incorporated too much noise for the evaluation, as seen in the error analysis of the relation classification, whereas the words in the dependency parser output demonstrated a greater performance. It is uncertain whether the features from KaBOB aided or hurt the performance in the relation classification, as the best algorithm (neural networks) performed seemingly just as well without them. Further testing is needed to determine if, for certain feature configuration, the addition of KaBOB-derived features results in a statistically significant performance improvement. We did not include results utilizing only KaBOB-derived features as part of the submission for this task, as they did not reach the top five performing methodologies. Their value lies, however, in the potential generalizations that can be made when using them, particularly after feature selection. These features derived from KaBOB could be used to estimate the probability of each of these relationships between any chemical-protein pair, based on their attributes in the knowledge-base. Feature selection algorithms and post-hoc analysis of the machine learning results would identify the aspects of the prior knowledge that were most helpful, and be used to generate hypotheses about generalizations (i.e. "chemicals with property X tend to down-regulate proteins that participate in molecular function Y").

There are various areas of improvement for this approach. We can refine our heuristics based on the error analysis data to address common misclassifications (i.e. "up-regulates" to "down-regulates", and vice-versa). It would also be very valuable to find the top shortest paths between a chemical entity and a protein entity in KaBOB, as it could directly yield how they are related, particularly if they intersect at a Reactome pathway step. Our KaBOB queries could also be expanded to include more information about chemicals and proteins that could be used as features. Performing cross-validation using the different approaches devised to determine proper statistical significance of the ones that seemed to perform best would be crucial, as well. Our classification could be extended by integrating features extracted using word embeddings, and training deep learning models using Recursive Neural Network (RNN) and Long Short-Term Memory (LSTM).

VII. Acknowledgments

We would like to acknowledge Elizabeth White, William A. Baumgartner Jr., Tiffany Callahan, Michael Bada, Anis Karimpour-Fard for their help and advice during this task.

VIII. References

1. K. M. Livingston, M. Bada, W. A. Baumgartner, and L. E. Hunter, "KaBOB: ontology-based semantic integration of biomedical databases," *BMC Bioinformatics*, vol. 16, p. 126, Apr. 2015.
2. ChemProt-3.0: A global chemical-biology diseases mapping. J. Kringelum, S.K. Kjærulff, T.I. Opera, S. Brunak, O. Lund, O. Taboureau.
3. Bada, M., Eckert, M., Evans, D., Garcia, K., Shipley, K., Sitnikov, D., Baumgartner Jr., W. A., Cohen, K. B., Verspoor, K., Blake, J. A., and Hunter, L. E. Concept Annotation in the CRAFT Corpus. BMC Bioinformatics. 2012 Jul 9;13:161. doi: 10.1186/1471-2105-13-161. [PubMed:22776079]
4. Jurafsky, D. and Martin, J.H. (2009) Ch. 22: Information Extraction, *Speech and Language Processing.* pp. 725-763.

CTCPI – Convolution Tree Kernel-based Chemical-Protein Interaction Detection

Neha Warikoo[125], Yung-Chun Chang[3], Po-Ting Lai[45] and Wen-Lian Hsu[5]
[1]Bioinformatics Program, Taiwan International Graduate Program, Institute of Information Science, Academia Sinica, Taipei, Taiwan,
[2]Institute of Biomedical Informatics, National Yang-Ming University, Taipei, Taiwan,
[3]Graduate Institute of Data Science, Taipei Medical University, Taipei, Taiwan,
[4]Department of Computer Science, National Tsing-Hua Univeristy, Hsinchu, Taiwan,
[5]Institute of Information Science, Academia Sinica, Taipei, Taiwan

***Abstract*:** In this paper we introduce a chemical-protein interaction detection system called CTCPI, which uses Convolution Tree Kernel to separate various levels of interaction mechanisms exhibited by chemicals while interacting with proteins/genes. Our system enlists a novel feature engineering method based on Algebraic Invariance to identify and consolidate distinct linguistic features for each class from the candidate sets and use these feature patterns as a screening function for generating the feature tree for SVM-CTK classifier. Our system achieved about 30.92% performance for chemical protein interaction class identification task.

***Keywords*:** Named Entity Recognition (NER), Algebraic Invariance, Convolution Tree Kernel (CTK)

I. Introduction

Biological interactions are vital in understanding the mechanism of action involved behind the physiological response to any drug. The nature of interaction can help us ascertain what category a drug belongs to, whether it inhibits an unwarranted gene regulation or works as a conjugate in some other reaction etc. There is plethora of published material with references to drug/chemical based interactions with various biological entities which can be used in subsequent researches to develop more effective drugs or be used in clinical practices for bettering patient care (7). BioCreative VI – ChemProt task is aimed at identifying relation types targeting "chemical compound-based interaction with proteins/genes" (CPI). The task is focused on identifying true interactions in proteins/genes stimulated or caused directly by chemical compounds/drugs and annotating the respective nature of interaction between the corresponding entities (4). As per the task we are to categorize these interactions into given 5 classes viz, Class 3 – Up Regulators, Class 4 – Down Regulators, Class 5 – Agonist, Class 6 – Antagonist and Class 9 – Substrate or Product Of.

II. Method

The Chemical Protein Interaction task is a multi-label classification task where named entity types for each abstract in the corpus data were provided by the task organizers. To deliver the task, we developed a 4-tier module system as shown in Figure 1. Tier1 is the Candidate Instance Generation System, which is responsible for pre-processing of input abstracts. This module entails screening for relation-oriented sentences, which are referred to as "Instances" in our system. This module is preceded by generic tasks of natural text preprocessing viz. Sentence Detection (Apache Open NLP) [1], Entity Class Labeling (In-built Module), and part-of-speech (POS) tagging (Genia Tagger)[2] (8-9). Hence generated POS-tagged sentences were drafted into candidate instances based on entity pairs (one chemical and one protein/gene pair mention per sentence per pairwise iteration) relabeling to indicate the primary Chemical and Protein pair. The verb implying the relation (proximal verb) was also assigned a prominent identifier as "Relation". It entails a non-basal form verb nearest to the current entity pair set. The use of this form can be explained by linguistic predominance in describing causal relations using a non-basal form verb in a smaller frame between the respective subject-object within the sentence.

The Tier2 of our system is based on a feature engineering method novel to the natural text domain. This module is dubbed as Algebraic Invariance-based Feature Engineering set. The module is responsible for generating a tree-based feature file where Invariance method is used for feature enrichment. In contrast to using the whole parse tree directly generated by Stanford parser (1), a set of stringent context feature patterns based on Invariance functional were used to set the upper and lower bounds for candidate instances accepted for tree generation. In addition, context patterns per instance were used for pruning and decorating the respective instance tree, thereby enhancing the feature context and bringing brevity to the features incorporated.

Our feature engineering approach is based on conjecture that different candidate instances show similarity in inference even if they are structurally diverse when relevant contexts are used as reference points. The objective is to demonstrate the invariance or lack of change in the nature of such descriptive sections from the text, and exploit this characteristic in generating more robust features while limiting the degree of

[1] http://opennlp.sourceforge.net/models-1.5/en-sent.bin
[2] http://www.nactem.ac.uk/tsujii/GENIA/tagger/geniatagger-3.0.2.tar.gz

evaluation function. The idea is heavily drawn on Algebraic Invariance to show that two separate sentences are similar in their inferential meaning if their invariant function does not vary (2). Such a function can be represented as follows:

$$I(q_{n0}...q_{0n}) \equiv \Delta^{W} * I(p_{n0}...p_{0n}) \qquad (1)$$

where *I(q)* and *I(p)* indicate the invariant function post and prior to transformation (T) , Δ is the determinant of the representational polynomial undergone transformation (T), and *W* is the invariant weight.

In order to restructure the invariance concept in a natural text paradigm, we assumed a homogenous polynomial function based on our prior selected three key referential labels viz. Entity1 (Chemical), Relation (Proximal Verb), and Entity2 (Protein/Gene) to project every instance in the Euclidian space. We limited our function to a second order polynomial based on each variable set as given below:

$$P(x,y) = p_{20}x^2 + p_{11}x^1y^1 + p_{02}y^2 \qquad (2)$$

where x and y are binary association variables representing "Entity1~Relation" and "Entity2~Relation" respectively. p_{20}, p_{11}, and p_{02} are coefficients of the polynomial evaluated by the maximum value from a seven-frame adjacency matrix vector for each of the corresponding variable pairs. This seven-frame adjacency matrix is calculated for each variable in the combination set and is based on an n-gram probabilistic model by shifting the window frame iteratively over the instance, moving reference label index from 1 through 7. We settled on 7-gram model as it tested positively with the development data. Per instance the cumulative n-gram score from union of similarity instances per variable over all candidate instances is taken as statistical significance score for the context pair set and is labeled as coefficient of the variable pair.

Our algorithm treats each candidate instance polynomial as a transformed version of all other instance polynomials. Using (1), if the invariant functional of the current candidate polynomial is equal to the invariant functional of other instance polynomials, then the current instance is considered similar to each of those instances, thereby reducing the dimensionality of screening space for pattern generation and keeping context-specific similarities.

Keren, D (4) developed a method for calculating invariance functional for homogenous polynomials. Using that methodology and based on (1) and (2), we determine our invariance functional as given below:

$$I(q_{n0}...q_{0n}) \equiv I(p_{n0}...p_{0n}) = \left(p_{20}^2 + \left(\frac{p_{11}^2}{2} \right) + p_{02}^2 \right) \qquad (3)$$

where *I(q)* and *I(p)* are the invariant functions for the transformed instance polynomial *Q(u,v)* and original instance polynomial *P(x,y),* respectively. p_{20}, p_{11}, and p_{02} are the coefficients of the original polynomial function *P(x,y)*. Every candidate instance is screened for the 3 key referential labels, and then corresponding coefficient values from the homogenous representation equation (2) are substituted in equation (3) to obtain the invariant function score $I(p)_k$ in

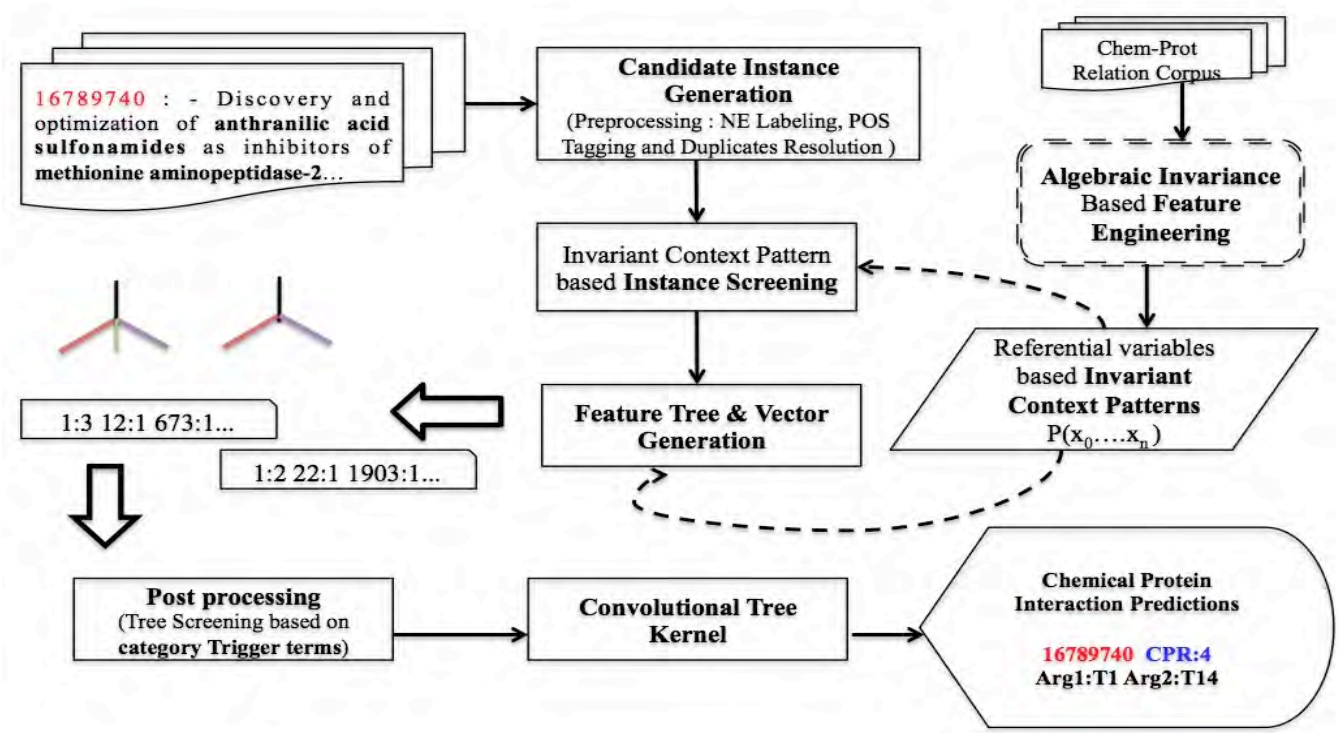


Figure 1 . CTCPI System Architecture

which k is the current candidate instance ID. The instances are sorted in descending order of their respective scores. If the invariant scores of two consecutive scores are approximately similar (Δ=1.00 and W~1, using (1)), then the instances are deemed as inferentially similar and thereby clustered together. Otherwise, the instances are diversified into separate groups. Instances within a group are aligned together to generate 7-frame size context pattern per key referential label. Consequently, we obtain per category specific screening patterns, which are used for feature enrichment in tree building. The context patterns generated are compared against the referential label-based frames from the candidate instances to screen the information content and relevance of the respective instances to be used for the classifier.

The hence generated pruned and decorated tree is subjected to Tier3 of our system module where further removal of noise from the candidate tree instances is attempted by using trigger words specific to each category. These trigger terms were identified based on statistically significant biological relational keywords obtained from the training corpus. The trigger terms are separated class wise and used to eliminate cross-referenced instances. Only filtered tree instances are allowed to be in the final feature file. The last segment i.e Tier4 of the tool is SVM classifier based on convolution tree kernel which is run on One vs One approach to segregate multiple interaction classes from each other. SVM-Light-TK-1.5[3] toolkit was used in both the learning and classification modules (5-6).

III. Experiments

Experimental SetUp

The dataset provided by the task organizers for separate phases varied in the abstract size. The training data extended to 1020 abstracts and 4157 interaction relations. The development data spanned about 612 abstracts with 2416 relations. Three out of five task submissions used a balanced clubbed model of training and development data while for the remaining two we employed a balanced training set instances for the model learning. We trained all the models on SVM-CTK. Other than using varied sizes of training data, we used variants of tree and vector features; by employing a combination model (tree + vector, #2), only tree-based model (#1,3,5) and only vector-based model (#4) for separate runs. The entity recognition was already provided with the respective corpuses, therefore our system only focused on chemical-based interaction with proteins/genes. The standard evaluation metrics of precision, recall and F1-score were used for calibrating the method efficiency.

Results and Discussion

Table 1 briefs the performance of our system for each of the runs submitted. A total of 5 runs were obtained for prediction on the test set. The system registered F1-Score of about 30.92% on the best run, with recall and precision measuring up to 32.71% and 29.32% respectively. The model with best run is based on a consolidated corpus using both training and development data, weighing heavily on tree-based features for classification in contrast vector features. Effectiveness of tree kernel in classification indicates that the class specific features introduced by our feature engineered tree pruning apparatus are relatively effective in enriching qualitative contribution of parse trees as opposed to using only vector features.

Run	Precision	Recall	F1-Score
#1	0.2446	0.3407	0.2847
#2	0.2563	0.3456	0.2943
#3	**0.2932**	**0.3271**	**0.3092**
#4	0.0729	0.0150	0.0249
#5	0.2587	0.3456	0.2959

IV. Conclusion

This paper introduces a system dedicated to identifying multi class chemical-protein interactions from pub-med abstracts. Multi-class classification makes this task challenging than usual binary classification tasks. In addition, there is also overlap in terms of interaction expression or key term references when it comes some interaction types, which brings in further ambiguity for classification. Our system was impactful to certain extent in segregating various interaction types, however an additional resource of trigger word features from curated literature database would be crucial in noise removal and better screening for interaction types.

References

1. Danqi Chen and Christopher D Manning. (2014) A Fast and Accurate Dependency Parser using Neural Networks. *Proceedings of EMNLP 2014.*
2. Leonard Eugene Dickson. (1914) Mathematical Monongraphs Algebraic Invariants, No.14. *John Wiley*, New York.
3. Daniel Keren. (1994) Using Symbolic Computation to Find Algebraic Invariants. *IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 16, No 11*, Nov 1994.
4. Krallinger, M., Rabal, O., Lourenço, A., et al. (2017) Information Retrieval and Text Mining Technologies for Chemistry. *Chemical Reviews*.
5. Alessandro Moschitti. (2006) Efficient Convolution Kernels for Dependency and Constituent Syntactic Trees. *Proceedings of the 17th European Conference on Machine Learning, Berlin, Germany, 2006.*
6. Alessandro Moschitti. (2004) A study on Convolution Kernels for Shallow Semantic Parsing. *Proceedings of the 42-th Conference on Association for Computational Linguistic (ACL-2004), Barcelona, Spain, 2004.*

[3] http://disi.unitn.it/moschitti/TK1.5-software/download.html

7. Rindflesch, T. C., Tanabe, L., Weinstein, J. N., & Hunter, L. (2000). EDGAR: extraction of drugs, genes and relations from the biomedical literature. *In Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing* (p. 517). NIH Public Access.
8. Yoshimasa Tsuruka, Yuka Tateishi, Jin-Dong Kim, Tomoko Ohta, John McNaught, Sophia Ananiadou, and Jun'ichi Tsujii. (2005) Developing a Robust Part-of-Speech Tagger for Biomedical Text, *Advances in Informatics - 10th Panhellenic Conference on Informatics, LNCS 3746, pp. 382-392, 2005*
9. Yoshimasa Tsuruoka and Jun'ichi Tsujii. (2005) Bidirectional Inference with the Easiest-First Strategy for Tagging Sequence Data, *Proceedings of HLT/EMNLP 2005, pp. 467-474*

Extracting Chemical-protein interactions via bi-directional long short-term memory network

Wei Wang[1], Xi Yang[1], Yuting Xing[1], Chengkun Wu[1], Zhuo Song[2]
[1]School of Computer Science, National University of Defense Technology, Changsha, China
[2]Genetalks Biotech. Co.,Ltd., Beijing, China

***Abstract*—Understanding chemical-protein interactions (CPI) has been of great importance to drug discovery, precision medicine and basic biomedical research. It is a time-consuming and laborious task to annotate CPIs from numerous unstructured texts. We can employ automated methods to improve the efficiency of this task. In this work, we propose a CPI extraction method based on the bi-directional long short-term memory network (a specific type of deep neural network), which does not require a complicated feature engineering procedure. Our key strategy is to break each sentence into fragments according to the position of the targeted entity pair and recombine them into chunks, which can help capture the structural knowledge hidden in the sentence. More specifically, our model consists of four network layers, including a feature layer, a Bi-LSTM layer, a pooling layer and a Softmax layer. Our results demonstrate that such a structure is beneficial for effective relation information.**

Keywords—Chemical-protein interaction; bi-directional long short-term memory network; structural knowledge;

I. INTRODUCTION

In clinical medicine, chemical drugs can act as therapeutic agents by targeting at some specific receptors and altering their structures, these receptors are usually some specific proteins, such as enzymes, oncogene products, anti-bodies, and etc. Therefore, understanding chemical and protein interactions (CPI) is of great importance to drug discovery, precision medicine and basic biomedical research. Biomedical researchers have studied a great amount of associations between chemicals and proteins, published their studies in the biomedical literature and added curated knowledge to some chemical-protein interaction databases, such as the protein data bank (https://www.rcsb.org/pdb/home/home.do) and the PDSP Ki Database (https://pdsp.unc.edu/databases/kidb.php). However, these databases are far from complete as it is time and labor-consuming to keep them up-to-date manually with the sharply growing volume of biomedical literature. Automated methods can greatly improve the efficiency of CPI extraction from unstructured texts.

Previously, various approaches were developed to address similar problems like DDIs (Drug-drug interactions) and PPIs (Protein-protein interactions) extraction. These approaches can be roughly divided into two categories: rule-based methods and machine learning methods. In general, the former approaches define a set of rules to capture various forms of expressing the relationship between two entities in texts. In (1), a set of syntactic rules and domain-specific lexical rules were applied in the identification of DDIs. Corney et al. (2) applied manually engineered templates that combine lexical and semantic information to identify PPIs. Besides manually crafted rules, it is also possible to generate rules automatically . For instance, Blasco et al. (3) proposed an automated method to summarize rules from a large amount of biomedical texts, which utilized Maximal Frequent Sequences (MFS) to discover patterns and such patterns have been proved to perform well on identifying sentences that contain DDIs. Generally speaking, rule engineering approaches are hard to scale up to large document collections due to various limitations.

Machine learning methods adopt statistical models to capture relational information via training on a set of training data. Several such methods have been proposed to extract DDIs from biomedical texts. (4) established the Turku event extraction system (TEES), and it supported detecting and identifying DDIs simultaneously by constructing a multi-class support vector machine (SVM). Liu et al. (5) introduced the convolutional neural network (CNN) into the DDI extraction task.

As for PPI extraction, feature-based methods and kernel-based methods are widely used. Feature-based methods focus on designing effective features including lexical, syntactic and semantic information. (6) used Maximum Entropy models to combine diverse lexical, syntactic and semantic features for PPI extraction. Kernel-based methods are even more effective for capturing syntactic structure information, which compute the structure similarity by kernel functions. Bunescu and Mooney (7) adopted a generalized substring kernel over a mixture of words and word classes to extract PPIs from biomedical corpora as well as semantic relations from the newswire corpora. Chowdhury et al. (8) investigated the effect of mildly extended dependency trees using an un-lexicalized partial tree kernel. Recently, deep learning techniques have achieved notable results in some PPI extraction tasks (9,10). However, to the best of our knowledge, deep learning is barely seen in CPI extraction yet.

In this work, we propose a bi-directional long short-term memory network (Bi-LSTM) based model to accomplish the task of CPI extraction without complicated feature engineering. A key strategy of our work is that we split the sentence into fragments and recombine them into chunks, expecting to capture the structure knowledge hidden in the sentence. Our model consists of four network layers, including a feature layer, a Bi-LSTM layer, a pooling layer and a

Softmax layer. In the feature layer, the sentence in each instance is split into three fragments according to the position of the target chemical entity and protein. We re-organize these three fragments into two chunks and represent them with word features and position features. Here the exact words are initialized with syntax word embedding and the position features are mapped into ten bit binary vectors. Subsequently, in the Bi-LSTM layer, two separate Bi-LSTM are equipped for each chunk with the scope of better learning relation information. After that, in the pooling layer, we employ piece max pooling rather than max pooling on the encoding sequence data obtained from the Bi-LSTM layer. Lastly, all results are concatenated together, and fed to the Softmax layer for CPI classification.

II. METHODS

In this work, we propose a bi-directional long short-term memory network based model for CPI extraction. Figure 1 shows the architecture of our model and the components are described in detail in the following parts.

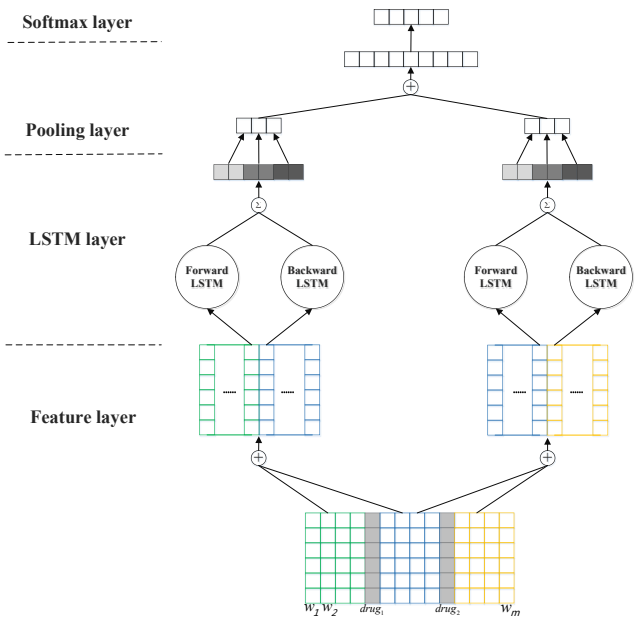


Figure 1 The framework of our model

A. Feature layer

Depending on the position of the two targeted entities, we split each instance into three fragments, named as $Fragment_1(F_1), Fragment_2(F_2), Fragment_3(F_3)$. F_1 denotes the fragment of a sentence in the front of the forward entity word, F_2 is the fragment between two targeted entities, and F_3 is the remaining fragment of the sentence behind the latter entity word. Subsequently, these three fragments are recombined into three chunks to represent the sentence. Here we combine F_1 and F_2 as $Chunk_1(\mathrm{Ch}_1)$, denoted as $\mathrm{Ch}_1 = F_1 + F_2$. Similar to Ch_1, F_2 and F_3 are combined into $Chunk_2(\mathrm{Ch}_2)$. In this way, we expect to capture the structure information of the sentence.

We follow earlier researches (11,12) to characterize each word in a sentence with word features and position features. Specifically, each word in a sentence is represented with three features: word (w), $\mathrm{Position}_1(P_1)$, and $\mathrm{Position}_2(P_2)$, where w is the exact word, P1 and P2 are the relative distances from the current word to two targeted entities (a negative distance means backwards).

B. Bi-LSTM layer

According to previous studies, Bi-LSTM has been proved to be an excellent model in processing long sequential data, especially for text data. Thus, we employ three Bi-LSTM model to encode each chunk constructed from the previous layer, aiming to better capture effective encoding information for relation extraction.

The Bi-LSTM model is equipped with two parallel LSTM layers, forward LSTM layer and backward LSTM layer. As theoretical analysis and experimental results show that the long sequence data exits long-term dependencies problem, the LSTM model emerged. Based on the recurrent neural network architecture, a new structure of the memory block is introduced into the LSTM model to alleviate the vanishing gradient problem. More precisely, the memory block consists of a memory cell (C_t) and three multiplicative gates, including the input gate (i_t), output gate (o_t) and forget gate (f_t). Respectively, the activation of the input gate multiplies the input to the cells, the output gate multiples the output to the net, and the forget gate multiplies the previous cell values. Figure 2 shows the detail structure of the memory block.

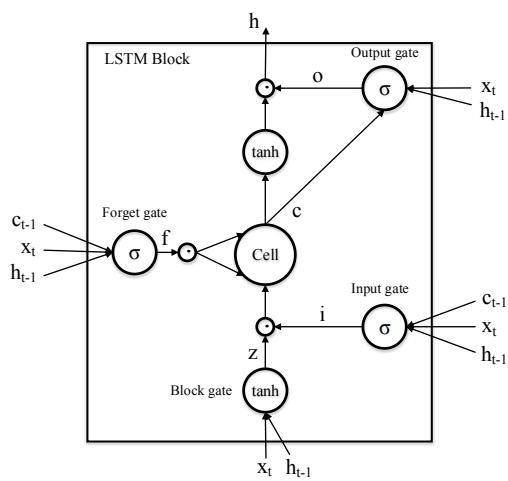


Figure 2 LSTM memory block

Consider x_i is the feature vector of the word, then the sequence data is denoted as $x_1, x_2, \ldots, x_i, x_m$, where m is the length of the sentence. Let h_{t-1} and c_{t-1} be the previous hidden and cell state of LSTM respectively. Thus, the computation of h_t and c_t would be:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i)$$

$$f_t = \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f\right)$$

$$z_t = \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_{c)}$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot z_t$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o)$$

$$h_t = o_t \cdot \tanh(c_t)$$

Here $W_{(.)}$ is the learning parameters for LSTM, o is the element-wise product, σ is the sigmoid activation function and b is the bias term.

As we employ the Bi-LSTM models, there would be two encoding results for the representation of the sequence data, h_t^f and h_t^b, which are produced by the forward LSTM layer and backward LSTM layer respectively. In this work, we add these two encoding results with the following operation:

$$z_t = h_t^f + h_t^b$$

C. Pooling layer

In this layer, we apply the piece max pooling, instead of max pooling, to get the optimal features from the entire sequence data, since the former method performs better than the latter method on obtaining more available information. Specifically, the encoding data learnt from Bi-LSTM layer is divided into several pieces with equal length. Let $z_k^1, z_k^2, \dots, z_k^t, z_k^l$ be the independent piece and $< v_t^1, v_t^2, \dots, v_t^d >_k$ be the vector of z_t^k, where k is the identifier of the piece, l is the length of the piece and d is the dimension. Then the result of piece max pooling would be:

$$z_k = < \max(v_t^1), \max(v_t^2), \dots, \max(v_t^d) >_k$$
$$z = z_1 + z_2 + z_k + \cdots + z_n$$

Where $\max(.)$ is to get the maximum value of each dimension wise. Subsequently, basing on the segmentation in the feature layer, we concatenate all the results as follow:

$$Z = z^{Ch_1} \oplus z^{Ch_2} \oplus z^{Ch_3}$$

D. Softmax layer

A softmax operation with dropout is set in this layer to give normalized probability score for each class. We use tanh as the activation function. The equations are given as follows:

$$h_s = \tanh(z)$$
$$p(y|x) = \mathrm{Softmax}(W^S h^s + b^s)$$

Where W is the softmax matrix and b is the bias term.

E. Model training

We utilized the *word2vec* tool to map each word into a numeric vector for word embedding, and the position features are mapped into a vector with 10 binary components. In addition, the weights and biases in our model are update by backpropagation through time. Specifically, we choose the cross entropy loss function and Adam's technique (13) with gradient clipping, parameter averaging and L2-regularization to train our model.

III. Experimental settings

1. In the CHEMPROT track, and to focus mainly on a subset of key relevant relation types, all the annotated CHEMPROT relations (CPRs) were grouped into 10 semantically related classes that do share some under-lying biological properties. Those groups are labeled as [CPR:1, CPR:2, … CPR:10] and a detailed description is shown in Table 1.

Table 1 The description of CHEMPROT relations

Group	Eval.	CHEMPROT relations belonging to this group
CPR:1	N	PART_OF
CPR:2	N	REGULATOR|DIRECT_REGULATOR|INDIRECT_REGULATOR
CPR:3	Y	UPREGULATOR|ACTICATOR|INDIRECT_UPREGULATOR
CPR:4	Y	DOWNREGULATOR|INHIBITOR|INDIRECT_DOWNREGULATOR
CPR:5	Y	AGONIST|AGONIST-ACTIVATOR|AGONIST-INHIBITOR
CPR:6	Y	ANTAGONIST
CPR:7	N	MODULATOR|MODULATOR-ACTIVATOR|MODULATOR-INHIBITOR
CPR:8	N	COFACTOR
CPR:9	Y	SUBSTRATE|PRODUCT_OF|SUBSTRATE_PRODUCT_OF
CPR:10	N	NOT

2. To keep the generalizability of our model, two entities in a pair are respectively replaced with "*ENTITY_1*" and "*ENTITY_2*". For instance, the CPI candidates in the sentence " The activities of UGTs 1A3, 1A8, 1A9, 2B4 and 2B7 were low, whereas *UGT1A1* and *UGT2B17* exhibited no *HFC* glucuronidation activity" are blinded as shown in Table 2.

Table 2 An example of entity blinding

CPI candidate	Entities blinding
(*HFC*, *UGT1A1*)	The activities of UGTs 1A3, 1A8, 1A9, 2B4 and 2B7 were low, whereas ***ENTITY_2*** and UGT2B17 exhibited no ***ENTITY_1*** glucuronidation activity.
(*HFC*, *UGT2B17*)	The activities of UGTs 1A3, 1A8, 1A9, 2B4 and 2B7 were low, whereas UGT1A1 and ***ENTITY_2*** exhibited no ***ENTITY_1*** glucuronidation activity.

3. Our solution is built based on Tensorflow[1] package using Python. Table 3 lists the hyper parameters used in the experiments.

Table 3 The hyper parameters of our model

Parameter	Description	Value
dw	Dimension of word embedding	100
dp	Dimension of position embedding	10
num	The number of hidden units	200
ρ	The ratio of dropout	0.7
l_2	The L2 regularization	0.001
l_a	The learning rate of Adam optimizer	0.001

4. We evaluated our model on the sample set with 339 instances. Our model outperforms the baseline in terms of recall and F-score, as listed in Table 4. We manually inspected a number of wrongly classified instances and analyzed several examples in Table 5. It turns out that long and complex sentences, especially those with clauses, are the ones that are prone to misclassification. This can be attributed to the limitation of bi-directional long short term-memory network on learning syntactic information from the extreme long and complex text in practice, although it can process the long

[1] https://www.tensorflow.org

sequential data in theory. More research needs to be done in order to address this kind of phenomenon.

Table 4 Performance comparison

	Precision	Recall	F-score
Our model	73.73	**66.95**	**70.16**
Baseline model	**85.21**	50.63	63.52

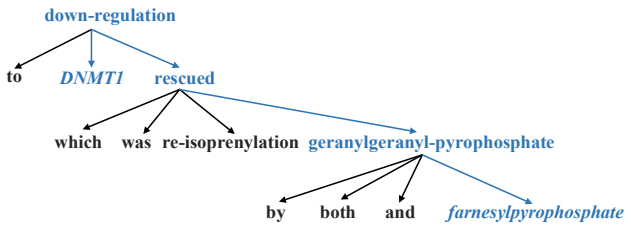


Figure 2 Part of dependency tree of instance S1

Table 5 Some examples of the classification errors

No.	Instance	Label	Prediction
S1	Further we found stimulation of FAS-expression as a result of epigenetic DNA demethylation that was due to down-regulation of ***DNMT1***, which was rescued by re-isoprenylation by both *geranylgeranyl-pyrophosphate* and ***farnesylpyrophosphate***.	CPR:3	CPR:4
S2	Our study shows that human TRPA1 is a target for apomorphine, suggesting that an activation of ***TRPA1*** might contribute to adverse side effects such as nausea and painful injections, which can occur during treatment with ***apomorphine***.	CPR:3	CPR:4
S3	Treatment of cells with BCNU to inhibit glutathione reductase (GR) enhanced the CpG-induced intracellular oxidation and decreased the ***GSH***/GSSG, with increased activation of NF-kappaB and a doubling in the CpG-induced production of ***IL-6*** and TNF-alpha.	CPR:4	CPR:3

ACKNOWLEDGMENT

This work is funded by the National Key Research and Development Program of China 2016YFB0200401.

REFERENCES

1. Segura Bedmar I. Application of information extraction techniques to pharmacological domain: extracting drug-drug interactions[J]. 2010.

2. Corney, D.P., Buxton, B.F., Langdon, W.B. and Jones, D.T. (2004) 'BioRAT: extracting biological information from full-length papers', Bioinformatics, Vol. 20, No. 17, pp.3206–3213.

3. García-Blasco S, Danger R, Rosso P. Drug-Drug interaction detection: A new approach based on maximal frequent sequences[J]. Procesamiento del Lenguaje Natural, 2010, 45: 263-266.

4. Björne J, Kaewphan S, Salakoski T. UTurku: drug named entity recognition and drug-drug interaction extraction using SVM classification and do-main knowledge[C]//Second Joint Conference on Lexical and Computational Semantics (* SEM). 2013, 2: 651-659.

5. Liu S, Tang B, Chen Q, et al. Drug-Drug Interaction Extraction via Convolutional Neural Networks[J]. Computational & Mathematical Methods in Medicine, 2016, 2016:1-8.

6. Xiao, J., Su, J., Zhou, G.D. and Tan, C. (2005) 'Protein–protein interaction extraction: a supervised learning approach', Proceedings of the Symposium on Semantic Mining in Biomedicine, European Bioinformatics Institute, Hinxton, UK, pp.51–59.

7. Bunescu R, Mooney R. Subsequence kernels for relation extraction. In: Proceedings of NIPS'2005. p. 171–8.

8. Chowdhury FM, Lavelli A, Moschitti A. A study on dependency tree kernels for automatic extraction of protein–protein interaction. In: Proceedings of BioNLP'2011. p. 124–33.

9. Hua, Lei, Quan, Chanqin. A Shortest Dependency Path Based Convolutional Neural Network for Protein-Protein Relation Extraction. BioMed Research International, 2016: 1-9

10. Hsieh, Yu-Lun & Chang, Yung-Chun & Chang, Nai-Wen & Hsu, Wen-Lian. (2017). Identifying Protein-protein Interactions in Biomedical Literature using Recurrent Neural Networks with Long Short-Term Memory. The 8th International Joint Conference on Natural Language Processing.2017.

11. Bobic T, Fluck J, Hofmannapitius M. SCAI: Extracting drug-drug interactions using a rich feature vector[J]. Relation Extraction, 2013.

12. Björne J, Kaewphan S, Salakoski T. UTurku: drug named entity recognition and drug-drug interaction extraction using SVM classification and domain knowledge[C]//Second Joint Conference on Lexical and Computational Semantics (* SEM). 2013, 2: 651-659.

13. Kingma D, Ba J. Adam: A method for stochastic optimization[J]. arXiv pre-print arXiv:1412.6980, 2014.

Combining Support Vector Machines and LSTM Networks for Chemical-Protein Relation Extraction

Farrokh Mehryary[1,2], Jari Björne[1,3], Tapio Salakoski[1,3], Filip Ginter[1]
1. Department of Future Technologies, University of Turku, Turku, Finland;
2. University of Turku Graduate School (UTUGS), University of Turku, Turku, Finland;
3. Turku Centre for Computer Science (TUCS), Turku, Finland

***Abstract*—We present the results of our participation in the BioCreative VI: Text mining chemical-protein interactions (CHEMPROT) track. The goal of this task is to promote the development and evaluation of systems capable of extracting relations between chemical compounds/drug and genes/proteins from biomedical literature. We participate with two systems: (1) an SVM system which relies on a rich set of features extracted from the parse graph and (2) an ensemble of neural networks that utilize LSTM networks and generate features along the shortest path of dependencies. We also combine the predictions from the two systems with the goal of increasing performance. On the development set, our system combination approach outperforms the two individual systems, achieving an F-score of 61.09 (according to the official evaluation metric). On the test set, our SVM system achieves the highest result for our submissions with an F-score of 60.99.**

Keywords — SVM; deep learning; ensemble learning; long short-term memory networks; LSTM; biomedical relation extraction;

I. Introduction

BioCreative VI Task 5 focuses on detection of statements of relations between chemical compounds/drugs and genes/proteins. The CHEMPROT corpus which provides such annotations is used as the training and test data in this task. The aim of the task is to promote the development of systems for extracting such relations for use in precision medicine, drug discovery and basic biomedical research[1].

BioCreative VI Task 5 follows the well-established approach of pairwise relation extraction in the field of biomedical text mining. Protein-protein interactions (PPI) were one of the extraction targets in the BioCreative II and BioCreative III challenges (1,2). The two Drug-Drug Interaction (DDI) shared tasks focused on the detection of adverse interactions between pairs of drugs (3,4). Considerable performance gains achieved using deep learning have recently been reported on the DDI Extraction 2013 corpus (5).

We approach the BioCreative VI Task 5 as a classification task where we classify each valid pair of entities as one of the annotated relation types or as a negative. We apply and compare two systems in this task, one based on artificial neural networks (ANN) and one on support vector machines (SVM). After optimizing these systems separately, we experiment with system combination, achieving increased performance on the development set. On the test set we note a considerable drop in the ANN performance, which requires further investigation.

II. Data

The CHEMPROT corpus is a pairwise relation dataset. All entities are given as known data for the participants, thus the task is to predict the relations for valid pairs of these entities. The relations are directed, always connecting a GENE type entity (gene or protein) to a CHEMICAL type entity. A large set of distinct types are used for annotating the relations, but these types are combined into 10 groups which are used as the actual classes for this task. Further, only five of these classes are taken into account in the task evaluation.

III. Methods

We develop two different systems capable of extracting relations between CHEMICAL and GENE entities. Our first system relies on a rich set of features and a linear Support Vector Machine (SVM) classifier. Features for this system are generated from the shortest dependency path connecting the two candidate entities in the sentence syntax dependency graph, from the linear order of tokens, a sentence bag of words and all dependency paths within 1–3 dependencies from the two entities. Our second system requires less feature engineering and is a deep learning-based system, utilizing an ensemble of three-channel long short-term memory networks. Features for this system are generated based on three information channels: words, part-of-speech (POS) tags and dependency type and word-adjacency edges, along the shortest path connecting the two entities. Finally, we combine predictions of the two systems to boost the F-score, using a simple algorithm that is optimized on the official development set. In this section we discuss the details of each approach.

A. Preprocessing

We use the TEES system (6) to run a preprocessing pipeline of tokenization, part-of-speech tagging, and parsing. We convert the CHEMPROT corpus into the Interaction XML

[1] http://www.biocreative.org/tasks/biocreative-vi/track-5/

format allowing it to be parsed by the TEES preprocessing system. We test different parses generated using the TEES preprocessor wrappers for the BLLIP, Stanford converter and SyntaxNet parser software (7-9). The default parsing pipeline in our experiments consists of BLLIP constituency parsing with the biomedical domain model of McClosky (10), followed by conversion to dependencies using the Stanford conversion tool (8). We test different variants of the Stanford Dependencies (SD) representation, with the *CCprocessed* variant being the default unless otherwise stated.

Since our systems work mainly based on the shortest dependency path connecting two candidate entities in a single sentence, we exclude any possible cross-sentence candidate pairs from the data. The training data incorporates ten different types of relations, 5 of them being evaluated in the task. We also define and add a ''negative'' type for the cases where no relation exists between the two candidate entities. Hence, we formulate this relation extraction task as an 11-class classification problem.

B. SVM-based system

The SVM-based system used in this work is the Turku Event Extraction System (6). The system is applied as-is, with no task-specific modifications. The TEES system uses the $SVM^{multiclass}$ software as the multiclass classifier implementation (11).

The TEES system relies on a rich feature representation. While most features are generated from the shortest path of dependencies, dependency chains outside this shortest path, bags of words and the linear order of tokens are also used in generating features, in an attempt to capture more of the sentence context outside the direct relation between the two entities of interest.

We test several different parses and ways of predicting the CHEMPROT corpus with TEES, but find that none of these improve performance over the default approach. In total we compare three ways of representing the corpus, 12 parses and the use of the DrugBank dataset (12) as additional features. For the three CHEMPROT corpus representations the TEES system is trained with either the default of all 10 classes, with the four non-evaluated classes merged into a single class or with the non-evaluated classes entirely removed. For parses, we try the BLLIP parser, with or without the McClosky biomodel and with all five types of Stanford conversion, as well as the SyntaxNet parser with or without its Universal Dependencies model.

C. Deep learning-based system

Our deep learning-based system (ANN for short) requires less feature engineering than the SVM-based system and generates the features along the shortest path that connects the two candidate entities in the syntactic parse graph. The shortest dependency path is known to contain most of the relevant words for expressing the relation between the two entities while excluding less relevant and uninformative words (13), hence many successful systems have been built around utilizing it (6,13-17).

The architecture of our deep learning-based system is centered around utilizing an ensemble of artificial neural networks, all having identical structure, but trained with different initial random weights. This is done to stabilize the variance in the measured performance, caused by the random initialization of the network weights.

Each neural network in the ensemble utilizes three separate long short-term memory networks (chain of LSTM units): for representing the sequence of words, the sequence of POS tags and the sequence of dependency types (DT, i.e., edges in the parse-graph) along the shortest path. We always traverse the path from the CHEMICAL entity to the GENE entity when generating features along the shortest path, regardless of the order of the entity mentions in the sentence. We notice this approach results in significantly better generalization for unseen data. Besides the existing dependency type edges in the parse graph, we add an artificial edge between any two adjacent words of the sentence (word-adjacency edges). As discussed by Quirk et. al (18), this approach mitigates the parsing errors and increases accuracy and robustness when the system is confronted with linguistic variation. We give the weight one to dependency type edges and the weight five to word-adjacency edges when searching for the shortest path in the graph.

The sequences of words/POS tags/dependency types are first mapped into sequences of their corresponding vector representations, i.e. embeddings, by three separate embedding lookup layers and then used as input for the LSTMs. For words, we use pre-trained word-embeddings provided by Pyysalo et al. (19), which have been trained on the texts of all PubMed titles and abstracts and PubMed Central Open Access (PMC OA) full text articles using the word2vec method (20). During the training of our system, word embeddings are fine-tuned while randomly initialized POS and dependency type embeddings are learnt from scratch. The outputs of the last LSTM unit of each of the three chains are concatenated and the resulting vector is fed to a fully connected hidden layer. The hidden layer finally connects to the decision layer, having an output dimensionality corresponding to the number of labels in the data set (plus one for the ''negative'' label) with softmax activation.

The network is trained on the official training data using the Nadam optimization algorithm. Applying a dropout (21) with the rate of 0.2 on the output of the first dense layer is the only explicit regularization method used. The training is stopped once the performance on the development set is no longer improving, measured using the official evaluation metric. Table I shows the comprehensive list of the hyperparameters used.

TABLE I. HYPERPARAMETERS OF THE NETWORKS

Hyperparameters	Values	
	Optimal value	*Tested values*
Word-adjacency edge weight	5	[3,4,5,6]
Word embedding dimensionality	200	pre-trained
POS embedding dimensionality	25	[25,50,75,100]
DT embedding dimensionality	25	[25,50,75,100]
Word LSTM, output dimensionality	300	[100,200,300,400]
POS tags LSTM, output dimensionality	200	[100,200,300,400]
DT LSTM, output dimensionality	200	[100,200,300,400]
Hidden layer, output dimensionality	200	[100,200,300,400, 500,600,700,800]
Activation functions	tanh	[tanh, sigmoid]
Dropout rate	0.2	[0 0.2 0.3 0.4 0.5]

[a.] The optimal and tested values for hyperparameters

To deal with the variance in the performance, we train an ensemble of 4 neural networks, all identical apart from the initial (random) weights. After training, each network predicts a set of confidences for each (development/test set) example. The final prediction for an example is generated by summing the confidences of all networks and choosing the label with the highest overall confidence.

D. System Combination

Our SVM and deep learning-based systems are trained with different sets of features. This is a potential case for investigating whether combining predictions of the two systems could help in achieving better performance for this task.

The system combination is implemented by merging the relation predictions from the two systems as either a union (OR) or an intersection (AND), and resolving overlapping predictions with conflicting types by using the classifier confidence scores. Since all entities are known data in this task, the predictions from the two systems can be aligned using pairs of gold standard entities.

If only one system predicts a relation for a given pair of entities, it is either included in (OR) or discarded from (AND) the combination. If both systems predict a relation, the relation with the higher confidence score is included in the combination. Both the SVM and ANN systems produce confidence scores in their own ranges. These ranges are normalized into the 0–1 interval for both systems, after which the normalized scores are compared. We experiment with combining all predictions, only positive predictions or only predictions for the evaluated classes and find that combining only positive predictions results in the best performance.

IV. RESULTS AND DISCUSSION

We conduct all of our experiments on the official development set using the official evaluation script provided by the organizers. Even though the data are annotated having ten different types of relations, the task only focuses on five of them by defining the official performance metric as the micro-averaged F-score of the five target classes. This is most likely due to the fact that there are much less training examples available in the data for the ignored classes. Table II shows the performance comparison of our different systems, evaluated on the development data.

TABLE II. PERFORMANCE OF THE SYSTEMS ON THE DEVELOPMENT SET

Evaluation on development set	Performance metrics		
	Precision	*Recall*	*F-Score*
SVM	64.55	54.72	59.23
ANN	61.90	55.01	58.25
SVM+ANN (OR, positive classes)	**58.45**	**63.99**	**61.09**
SVM+ANN (AND, positive classes)	75.42	48.14	58.77
SVM+ANN (OR, all classes)	65.82	55.55	60.25
SVM+ANN (AND, all classes)	65.82	55.55	60.25
SVM+ANN (OR, eval classes)	56.47	65.07	60.46
SVM+ANN (AND, eval classes)	79.28	45.78	58.04

As Table II shows, both the SVM and deep learning-based (ANN) systems have very similar performance on the task, with the SVM having an F-score 1pp above the ANN. This might be due to the fact that the ANN solely relies on the words and edges seen on the shortest path and we suspect that in many cases, the *trigger word* (i.e., a token or sequence of tokens which expresses the actual relation between the two candidate entities) might be absent from this path. Consequently, the ANN might not get the chance to see this information, whereas the SVM system generates features based on all tokens and dependencies near the two entities, as well as those on the shortest path connecting them. The best SVM performance is achieved with the TEES default settings, without using the DrugBank features, using the BLLIP+biomodel+CCProcessed parsing approach and including all ten CHEMPROT relation types in the training data.

For both systems, recall is considerably lower than precision (for instance, recall is 10pp below precision for the SVM). Using the OR operation in system combination considerably improves the recall (~9pp) while causing a comparatively lower drop in precision, leading to an approximately 1–1.5pp increase in the resulting F-score. We observe that discarding negative predictions and building the combination from all 10 positive classes results in the highest performance on the development set.

For predicting the test set, we combine the training and development data when training the SVM system. This is a quite common approach when using classifiers such as SVMs. However, training the neural networks on the combined data for the *optimal* number of epochs (found during the optimization) might lead to under/over-fitting, because more/less training epochs might be needed. Finding the optimal number of epochs for training the network on the combined data is challenging. In this task, participating teams were allowed to submit up to 5 different test set predictions. Hence, we submitted two sets of ANN predictions: (1) predictions of the ensemble of networks that are trained for 3 epochs (the optimal number found in optimization), (2) predictions of the ensemble when the networks are trained for 4 epochs. We also combined these two sets of predictions with the SVM system predictions (using our system combination approach), resulting in a total of five sets of test set predictions. Table III shows the official results for our submissions on the test set, as calculated by the task organizers.

TABLE III. PERFORMANCE OF THE SYSTEMS ON THE TEST SET

Evaluation on test set	**Performance metrics**		
	Precision	***Recall***	***F-Score***
SVM	66.08	56.62	60.99
ANN (3 epochs)	63.73	44.62	52.49
ANN (4 epochs)	63.37	43.87	51.85
SVM+ANN (3-epochs)	61.05	60.06	60.55
SVM+ANN (4-epochs)	60.88	59.89	60.38

As Table III shows, compared to the development set results, our SVM system has approximately the same level of performance on the test set, achieving an F-score of 60.99, with a similar imbalance between precision (66.88) and recall (56.62). However, for the ANN submissions we notice a significant drop in recall (~11pp) with a small increase in precision (~1pp), leading to an F-score of 52.49 (in the case the networks are trained for 3 epochs) or 51.85 (when the networks are trained for 4 epochs), which is about 6pp below the F-score seen on the development set. As a direct result, none of the two system combination approaches have been able to produce a result better than the SVM system alone.

Since at the time of manuscript submission test set labels have not yet been published, we cannot perform a comprehensive analysis on the results of the ANN system. One potential explanation could be related to (over) training the networks with the combined data, which together with the small mini-batch size may have led to some overfitting. Particularly, we notice a considerable drop in the training-loss between the second and third epochs, which is approximately double the difference of the loss between the third and fourth epochs. This might indicate that overfitting would have occurred when training the networks for the third epoch.

Since the organizers are going to publish the test set labels, as our first additional experiment we would like to investigate what would be the performance of the ANN system on the test set using the version of the networks trained solely on the training set and optimized on the development set.

V. CONCLUSIONS AND FUTURE WORK

We participated in the CHEMPROT track of the BioCreative VI shared task with two different systems. Our SVM system relies on a rich set of features, extracted from the sentence parse graph, whereas our deep learning-based system requires less feature engineering and is an ensemble of three-channel LSTM networks. Features for this system are generated based on the shortest path which connects the two candidate entities.

Experiments on the development set show that combining the predictions of the two systems can lead to overall performance higher than that of either of the two systems alone. While the SVM system performs equally well on the development and test sets, the ANN system performance is considerably lower on the test set. We aim to investigate the possible reasons behind this result as soon as the test set labels are published.

The SVM system uses many features in addition to those extracted from the shortest path of dependencies. As future work, we would like to study the effect of each of these feature types on the performance for this task, as well as investigate efficient ways to incorporate and utilize such features with our deep learning -based approach.

ACKNOWLEDGMENTS

This work was supported by an ATT Tieto käyttöön grant. Computational resources were provided by CSC - IT Center for Science Ltd., Espoo, Finland.

REFERENCES

1. Krallinger, M., Leitner, F., Rodriguez-Penagos, C., Valencia, A., et al. (2008). Overview of the protein-protein interaction annotation extraction task of BioCreative II. Genome Biology, 9(Suppl 2):S4.

2. Krallinger, M., Vazquez, M., Leitner, F., Salgado, D., et al. (2011). The Protein-Protein Interaction tasks of BioCreative III: classi cation/ranking of articles and linking bio-ontology concepts to full text. BMC Bioinformatics, 12(Suppl 8):S3.

3. Segura-Bedmar, I., Martínez, P., and Herrero Zazo, M. (2013). SemEval-2013 Task 9: Extraction of Drug-Drug Interactions from Biomedical Texts (DDIExtraction 2013). In Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the

Seventh International Workshop on Semantic Evaluation (SemEval 2013), pages 341–350, Atlanta, Georgia, USA. Association for Computational Linguistics.

4. Segura-Bedmar, I., Martínez, P., and Sánchez-Cisneros, D. (2011). The 1st DDIExtraction-2011 challenge task: Extraction of Drug-Drug Interactions from biomedical texts. In Proceedings of the 1st Challenge Task on Drug-Drug Interaction Extraction 2011: 7 Sep 2011; Huelva, Spain, pages 1–9.

5. Liu, S., Chen, K., Chen,Q., and Tang, B. (2016) Dependency-based convolutional neural network for drug-drug interaction extraction. In Proceedings of IEEE-BIBM, pp.1074-1080.

6. Björne, J. (2014) Biomedical Event Extraction with Machine Learning. Ph.D. thesis, University of Turku.

7. Charniak, E., and Johnson, M. (2005) Coarse-to-fine N-best parsing and maxent discriminative reranking. In proceedings of ACL, pp.173:180.

8. de Marneffe, M.-C., MacCartney, B. and Manning, C. D. (2006) Generating Typed Dependency Parses from Phrase Structure Parses. In proceedings of the LREC-2006, pp.449:454.

9. Andor D., Alberti C., Weiss D., Severyn A., Presta A., Ganchev K., Petrov S., and Collins M. (2016) Globally Normalized Transition-Based Neural Networks. CoRR abs/1603.06042. http://arxiv.org/abs/1603.06042.

10. McClosky, D. (2010) Any Domain Parsing: Automatic Domain Adaptation for Natural Language Parsing. Ph.D. thesis.

11. I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. (2005) Large Margin Methods for Structured and Interdependent Output Variables. Journal of Machine Learning Research (JMLR) 6(Sep), pp.1453–1484.

12. Knox C., Law V., Jewison T., et al. (2011) Drugbank 3.0: a comprehensive resource for omics research on drugs. Nucleic Acids Research, 39(Database-Issue):1035–1041.

13. Mehryary, F., Björne, J., Pyysalo, S., Salakoski, T. and Ginter, F. (2016) Deep Learning with Minimal Training Data: TurkuNLP Entry in the BioNLP Shared Task 2016. In proceedings of the 4th BioNLP Shared Task Workshop, BioNLP 2016, pp.71-81.

14. Mehryary, F., Hakala, K., Kaewphan, S., Björne, J., Salakoski, T. and Ginter, F. (2017) End-to-End System for Bacteria Habitat Extraction. In proceedings of BioNLP 2017, pp.80-90.

15. Cai, R., Wang, H., and Zhang, X. (2016) Bidirectional Recurrent Convolutional Neural Network for Relation Classification. In proceedings of ACL, pp.756-765.

16. Xu, Y., Mou, L., Li, G., Chen, Y., Peng, H. & Jin, Z. (2015) Classifying Relations via Long Short Term Memory Networks along Shortest Dependency Paths. In Proceedings of EMNLP, pp.1785-1794.

17. Bunescu, R. C.and and Mooney, R. J. (2005) A shortest path dependency kernel for relation extraction. In proceedings of HLT-EMNLP, pp.724–731.

18. Quirk, C., and Poon, H. (2017) Distant Supervision for Relation Extraction beyond the Sentence Boundary. In Proceedings of EACL, pp.1171-1182.

19. Pyysalo, S., Ginter, F., Moen, H., Salakoski, T. and Ananiadou, S. (2013) Distributional Semantics Resources for Biomedical Text Processing. In proceedings of LBM 2013. pp. 39-44.

20. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013) Distributed Representations of Words and Phrases and Their Compositionality. In proceedings of NIPS, pp.3111-3119.

21. Srivastava, N., Hinton, G.E., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R. (2014) Dropout: a simple way to prevent neural networks from overfitting. *Machine Learning Research.*, **15**, 1929-1958.

Improving the learning of chemical-protein interactions from literature using transfer learning and word embeddings.

P. Corbett, J. Boyle
Data Science Group, Technology Dept., The Royal Society of Chemistry

***Abstract*—In this paper we explore the application of artificial neural network ("deep learning") methods to the problem of detecting chemical-protein interactions in PubMed abstracts. We present here a system using multiple Long Short Term Memory layers to analyse candidate interactions, to determine whether there is a relation, and which type. A particular feature of our system is the use of unlabeled data, both to pre-train word embeddings, and also pre-train LSTM layers in the neural network. On the BioCreative VI CHEMPROT test corpus, our system achieves an F score of 61.51% (56.10% precision, 67.84% recall).**

Keywords—chemicals, proteins, text mining, relationship extraction, deep learning, LSTM.

I. Introduction

The BioCreative VI CHEMPROT task concerns the detection of mentions of interactions between chemical compounds/drugs and genes/proteins. Whereas there has been much work in the detection of chemical and gene/protein named entities, and in the recognition of some interactions involving these entities (1), there has been little work (2,3) so far on chemical-protein interactions.

The CHEMPROT corpus consists of PubMed abstracts manually annotated with chemical compound mentions, gene/protein mentions, and chemical compound-protein relations. Each relation annotation has one chemical compound mention, one gene/protein mention, and a relationship type. There are 22 relationship types, collected into ten groups, of which five groups are used in the CHEMPROT task (relations from the other five groups are discarded). The five relation groups are upregulator/activator (CPR:3), downregulator/inhibitor (CPR:4), agonist (CPR:5), antagonist (CPR:6), and substrate/product (CPR:9). The annotated abstracts are provided in three groups – 1020 training abstracts, 612 development abstracts and 3399 test abstracts.

We decided to examine the use of the neural network techniques known as "deep learning" (4) – in particular, using the recurrent neural network components known as Long Short Term Memory (LSTM) layers. One advantage of these deep learning methods is that they provide methods for exploiting unlabelled data. One method that we explored was the use of word embeddings – *n*-dimension vector representations of words – pretrained on large, relevant corpora. A second method was to exploit transfer learning – to build a neural network that can be trained on some task with just the use of an unlabelled corpus, train it, and then re-use some of the trained components from that network in the task of interest. One possible training task is to predict the next (or previous) word in a sentence or paragraph given all of the preceding (or following) words.

II. Methods

A. Resources

We used various external components in our system. The software components include the deep learning toolkit keras (5) - using tensorflow as the back end, python 3.6.1, and the tokeniser chemtok, as implemented in the chemical named entity recognition system ChemListem (6).

To prepare pre-trained word embeddings, we used the Stanford GloVe software (as checked out from version control July 3 2017). GloVe offers both some public pre-trained embeddings (based on Wikipedia and Common Crawl), and also the software to compile your own – in previous work (6) we had success with the public embeddings, but here, we compiled our own. To create these embeddings we prepared three corpora – the full texts of patents, consisting of patents with CPC codes A61K31 or A61P, from 2006 to November 2016, the full text of chemistry journal papers, consisting of papers published by the Royal Society of Chemistry from 2000 to end of 2016, and the titles and abstracts from PubMed records from 1809 to the end of 2015.

To make the initial embeddings, we extracted the text from the documents in the three corpora, tokenised it, outputting whitespace-separated tokens as one large text file. The contents of one document was separated from the next using lines consisting of "$GLOVEDUMMY " repeated 16 times. We used the GloVe software to extract 300-dimensional vectors from the file, using the window size = 15, xmax = 100.

We also prepared a file for transfer learning, taking the titles and abstracts from PubMed as mentioned above. The file consisted of on paragraph (usually a title or abstract) per line, in a random order. The file had approximately 24 million lines.

B. Neural Network

The neural network system consisted of two neural networks – the "pretraining" network and the "recognition" network - with some components shared by both networks, and

other components being used by only one network or the other. The training procedure consisted of a series of epochs, the first five of which were divided into two phases – one (phase 1) to train the "pretraining" network, one (phase 2) to train the "recognition" network. All subsequent epochs after the fifth omitted phase 1 and ran phase 2 only. At the end of each epoch, the system was evaluated using the development abstracts, and an answer file was produced using the test abstracts. The epoch that gave the best F score in the evaluation phase – in the run submitted, the 33rd epoch – was selected, and the answer file from that was submitted for official evaluation.

Each run of phase 1 was divided into 25 sub epochs. In each sub-epoch, 12000 lines of the PubMed file were read in, tokenised, and sorted into batches of 32 lines each, grouping the smallest 32 lines (by number of tokens) into one batch, the next smallest 32 into another batch, etc. Within each batch, lines that are shorter (in terms of number of tokens) than the maximum length were padded with special padding tokens. The system was trained on the batches in a random order.

For each line, a token sequence was generated, consisting of an integer representing the index of each token in a token dictionary, with a special value for unknown tokens. From this a "substituted" sequence – where each token has a 0.5 chance of being replaced by a token randomly sampled from the lines read in that sub-epoch – was generated.

The inputs to the "pretraining" network consisted of the token sequence (input **i1**), the "substituted" sequence shifted one token to the right (input **i2**) (starting with padding), and the substituted sequence shifted one token to the left (ending with padding) (input **i3**). There were two outputs (**d2** and **d4**), one for each of the substituted shifted sequences, consisting of a sequence of numbers – 1 if the token in the substituted sequence is from the original sequence, 0 if it was randomly selected.

The network consisted of various layers, as shown in Table I. In all cases the number of output neurons is per token. The three embedding layers all shared the same embedding tensor. All LSTM layers were trained with a dropout and recurrent_dropout parameter of 0.5, and with return_sequences set to True.

TABLE I. LAYERS IN PRETRAINING NETWORK

Layer	Type	Input(s)	Number of output neurons	Notes
e1	Embedding	**i1**	300	
e2	Embedding	**i2**	300	
e3	Embedding	**i3**	300	
l1	LSTM	**e1**	300	
l2	LSTM	**e1**	300	Reversed
c1	concatenate	**l1**, **e2**	600	
c2	concatenate	**l2**, **e3**	600	
d1	TimeDistributed Dense	**c1**	300	activation is relu
d2	TimeDistributed Dense	**d1**	1	activation is sigmoid.
d3	TimeDistributed Dense	**c2**	300	activation is relu
d4	TimeDistributed Dense	**d3**	1	activation is sigmoid.

The "pretraining" network was trained using RMSProp optimizer, with the binary cross-entropy loss function.

In the second phase, chemical-protein relations were detected and classified. Each epoch consisted of a single pass through the training corpus to train the network, a single pass through the development corpus to evaluate the current state of the system, and a single pass through the test corpus to generate an answer file for submission.

In each pass, for each abstract, all possible chemical-protein pairs were found. Those pairs where the first token of the first entity was 60 or fewer tokens from the last token of the last entity were selected. A subsequence of tokens from the abstract was then taken, starting from 5 tokens before the first entity to 5 tokens after the last entity. The tokens for the chemical entity were replaced with "$CHEMICAL" and those for the protein entity were replaced with "$PROTEIN" – those appearing in both entities were replaces with "$BOTH". The token sequence was then converted to an integer sequence, in the same manner as the pretraining sequences were processed. Additional input sequences for each pair were also generated, consisting of an array of binary features for each token in the subsequence. One input sequence (input **i4**) consists of information about the entities in the abstract, regardless of whether they participated in the relation in question – these were features to say whether the token is in, at the start of, at the end of, overlapping the start of or overlapping the end of any chemical or protein entity. Another input sequence (input **i5**) consists of binary features to say whether the token is a part of the protein entity in question, and whether the token is a part of the chemical entity in question.

The output for the network (**d5**) was an array of 6 binary features, encoding whether and which relation exists between the two entities.

The network consisted of various layers, as shown in Table II. The number of output layers is per token, except for layers **p1** and **d5**, where it is the total number overall. The layers **e1**, **l1** and **l2** are shared with the pretraining network. Again, all LSTM layers were trained with a dropout and recurrent_dropout parameter of 0.5, and with return_sequences set to True.

TABLE II. LAYERS IN RECOGNITION NETWORK

Layer	Type	Input(s)	Number of output neurons	Notes
e1	Embedding	**i1**	300	
l1	LSTM	**e1**	300	
l2	LSTM	**e1**	300	Reversed

v1	Conv1D	**i3**	48	width=3, activation is relu
v2	Conv1D	**i4**	6	width=3, activation is relu
c3	concatenate	**l1**, **l2**, **v1**, **v2**	652	
l3	Bidirectional LSTM	**c3**	128 per direction, total 256	
p1	GlobalMaxPooling1D	**l3**	256	
d5	Dense	**p1**	6	activation is softmax

The network was trained using RMSProp, with the mean squared error loss function. During training, the candidate relationships were grouped into batches by length, if necessary padding the sequences to make the length of all the sequences in a batch uniform. The batches were then used for training in a random order.

C. ADDITIONAL EXPERIMENTS

To assess the importance of the techniques that used the unlabelled data, two additional runs were performed. One run ("No Phase 1") omitted the phase 1 from training. The second ("Random") also omitted phase 1, and also used randomly-initialised embeddings rather than the GloVe-trained ones. On the No Phase 1 run, the best epoch was the 15th epoch, and on the Random run, the best epoch was the 17th epoch.

III. RESULTS AND DISCUSSION

Table III shows the results from the task:

TABLE III. RESULTS

Corpus	*Precision*	*Recall*	*F*
Development	56.52%	70.42%	62.71%
Test	56.10%	67.84%	61.41%

The F of less than 63% indicates that there is considerable room for improvement on this task. This is the first time that BioCreative has tackled a chemical-protein interaction task – however, in the past it has considered chemical-disease relations (getting a maximum F score of 57.03%) and protein-protein interactions (getting a maximum F of 55%). These relationship-mining tasks appear to be harder than named entity extraction tasks, where F scores in excess of 80% are routine and F scores above 90% are not unknown (10). There appears to have been a slight loss of performance between the development and test – it is possible that this is because the gains from selecting the best epoch did not generalize well.

Table IV shows a confusion matrix for the development data.

TABLE IV. CONFUSION MATRIX FOR DEVELOPMENT DATA

Actual	Predicted					
	NONE	***CPR:3***	***CPR:4***	***CPR:5***	***CPR:6***	***CPR:9***
NONE	26196	214	413	75	76	351
CPR:3	163	287	82	4	4	9
CPR:4	159	24	896	0	5	6
CPR:5	23	0	0	89	4	0
CPR:6	28	1	7	3	160	0
CPR:9	166	4	16	0	2	258

The major source of error seems to be non-relations being mistaken for relations and vice versa. There is something of a problem with upregulation (CPR:4) being mistaken for downregulation (CPR:3) but this is not the biggest problem for either relationship class.

TABLE V. DEVELOPMENT DATA RESULTS BY RELATIONSHIP CLASS

Class	Precision	Recall	F
CPR:3	54.16%	52.28%	53.20%
CPR:4	63.37%	82.20%	71.57%
CPR:5	52.04%	76.72%	62.02%
CPR:6	63.75%	80.40%	71.11%
CPR:9	41.35%	57.85%	48.22%

There is considerable variation in how well these entities are recognised – CPR:4 (downregulator/inhibitor) and CPR:6 (antagonist) are well-recognised, CPR:3 (upregulator) and CPR:9 (substrate/product) are poorly recognised. The F scores do not appear to be correlated with the number of mentions in the corpus.

TABLE VI. RESULTS ON DEVELOPMENT

Run	*Precision*	*Recall*	*F*
Full	56.52%	70.42%	62.71%
No Phase 1	62.97%	57.25%	59.97%
Random	45.05%	50.66%	47.70%

Table VI shows the results of re-running the system, progressively disabling parts of the system that make use of unlabelled data. The Phase 1 training of the lower LSTMs is shown to improve performance by 2.7 percentage points; the GloVe-trained embeddings are worth 12.2 percentage points.

IV. CONCLUSION

Methods based on "deep learning" recurrent neural networks can be used to detect relationships between chemicals and protein, with results comparable with those observed in other biomedical relationship extraction tasks. The deep learning structure allows the use of large amounts of unlabelled

text to boost performance, especially via the use of pre-trained word embeddings.

The source code for our system is available on-line, as a part of the distribution for ChemListem, at https://bitbucket.org/rscapplications/chemlistem.

ACKNOWLEDGMENT

We would like to thank Colin Batchelor, Aileen Day, Nicholas Bailey and Jeff White for valuable discussions.

REFERENCES

1. Krallinger,M., Rabal,O., Lourenço,A., Oyarzabal,J. and Valencia,A. (2017) Information Retrieval and Text Mining Technologies for Chemistry. *Chem. Rev.*, **117**, 7673-7761.

2. Craven,M. and Kumlien,J.. (1999) Constructing biological knowledge bases by extracting information from text sources. In *ISMB*, **1999**, 77-86.

3. Rindflesch, T. C., Tanabe, L., Weinstein, J. N., and Hunter, L. (2000). EDGAR: extraction of drugs, genes and relations from the biomedical literature. In *Pacific Symposium on Biocomputing,* 517.

4. LeCun,Y., Bengio,Y. and Hinton,G. (2015) Deep Learning. *Nature* **521** 436-444.

5. Chollet, F. (2015) "Keras" https://github.com/fchollet/keras.

6. Corbett, P, Boyle, J. Chemlistem - chemical named entity recognition using recurrent neural networks. In *Proceedings of the BioCreative V.5 Challenge Evaluation Workshop*, 61-68.

7. Pennington,J., Socher,R. and Manning,C. (2014) GloVe: Global Vectors for Word Representation.

8. Wei,C.-H., Peng,Y., Leaman,R., Davis,A.P., Mattingly,C.J., Li,J., Wiegers, T.C. and Lu,Z. (2015) Overview of the BioCreative V Chemical Disease Relation (CDR) Task. In *Proceedings of the Fifth BioCreative Challenge Evaluation Workshop*, 154-166.

9. Kralliner, M., Vazquez, M., Leitner, F. And Valencia, A. (2011) Results of the BioCreative III Interaction Method Task. In *Proceedings of BioCreative III Workshop*, 5-12.

10. Pérez Pérez,M., Rabal,O., Pérez Rodríguez,G., Vazquez,M., Fdez-Riverola,F., Oyarzabal,J., Valencia,A., Lourenco,A. and Krallinger,M. (2017) Evaluation of chemical and gene/protein entity recognition systems at BioCreative V.5: the CEMP and GPRO patents tracks., in *Proceedings of the BioCreative V.5 Challenge Evaluation Workshop*, 3-11.

CNN-based Chemical-Protein Interactions Classification

Atakan Yüksel[1], Hakime Öztürk[1], Elif Ozkirimli[2], and Arzucan Özgür[1]
1: Department of Computer Engineering, Boğaziçi University, İstanbul, Turkey
2: Department of Chemical Engineering, Boğaziçi University, İstanbul, Turkey
email: atayuksel@outlook.com, {hakime.ozturk, elif.ozkirimli, arzucan.ozgur}@boun.edu.tr

***Abstract*— In this study, we present a Convolutional Neural Network (CNN) based model for the extraction and classification of different groups of interactions between chemical and protein pairs for the Text Mining Chemical-Protein Interactions (CHEMPROT) task of BioCreative VI . We used word-embeddings and distance embeddings to represent a potential relation. Our system obtained 0.68 F-measure on the CHEMPROT development set.**

Keywords—CNN; relation classification; chemical-protein interactions; word-embeddings; deep learning

I. Introduction

Identification of the biomedical interactions (relations) constitutes an important task as the number of articles indexed in Pubmed, which is the main source for the biomedical literature, continues to grow rapidly. The information load in the literature leads to the construction of many independent databases that store different interaction types such as the ones among proteins, protein-ligand, gene-disease or drugs. Keeping these databases up-to-date requires a major manual effort considering the large amount of information, therefore the need for an automated system to extract the important information from the text is urgent.

Previous studies on biomedical text mining mostly addressed the problems of extracting the interactions among proteins and/or genes, and drugs from the biomedical literature [1, 2, 3, 4] whereas interactions between proteins and ligands has not been well studied yet. As a result, there has been a lack of an annotated corpus that could be used to evaluate the developed extraction models. BioCreative VI provided a manually annotated chemical-protein interaction corpus, CHEMPROT, which has labels for (i) chemical and protein/gene names and (ii) type of the binary relationships among these entities.

Among the few existing studies, Chang et al. developed a rule-based approach to extract protein-ligand binding affinity data from the literature [5]. Their approach is based on manually designed patterns that make use of the surface forms of the sentences (i.e., sequences of words). However, the design of the patterns is a non-trivial task considering there are many different and complex ways to express the same information. In a recent study, Random Forest algorithm along with dependency-based analyses was utilized to extract GPCR-ligand interactions from the literature [6].

Following the striking performance in computer vision field, Convolutional Neural Networks (CNNs) have been adopted by many research areas including text mining. CNNs have been successfully applied to named entity recognition, relation extraction and classification tasks [3, 7, 8, 9, 10]. CNNs are especially better at recognizing patterns in sequences with the help of filter mechanisms that learn different local features and the pooling layer that combines local features into a global one.

CHEMPROT corpus provided by BioCreative VI organizers contains manually annotated chemical and protein/gene entities for each abstract. In this study, we adopt CNN model to extract chemical-protein relations from biomedical abstracts and to classify the relation into the correct interaction group.

II. METHODS

A. Data set

We used the CHEMPROT corpus, which contains abstracts with annotated entities for training (1020), development (612) and test (3399) sets. Interactions in the CHEMPROT corpus are grouped into total ten classes of biologically chemical-protein relations (CPR) only five of which are included in the evaluation. Table I summarizes the type of CPRs in detail (http://www.biocreative.org/tasks/biocreative-vi/track-5/).

TABLE I. CPRs in CHEMPROT Corpus

CHEMPROT relations	***Group***	***Evaluation***
PART_OF	CPR:1	N[a]
REGULATOR|DIRECT_REGULATOR|INDIRECT_REGULATOR	CPR:2	N
UPREGULATOR|ACTIVATOR|INDIRECT_UPREGULATOR	CPR:3	Y[b]
DOWNREGULATOR|INHIBITOR|INDIRECT_DOWNREGULATOR	CPR:4	Y
AGONIST|AGONIST-ACTIVATOR|AGONIST-INHIBITOR	CPR:5	Y
ANTAGONIST	CPR:6	Y
MODULATOR|MODULATOR-	CPR:7	N

CHEMPROT relations	*Group*	*Evaluation*
ACTIVATOR|MODULATOR-INHIBITOR		
COFACTOR	CPR:8	N
SUBSTRATE|PRODUCT_OF|SUBSTRATE_PRODUCT_OF	CPR:9	Y
NOT	CPR:10	N

[a,b:] Y (Yes), N (No) included in the evaluation

B. Input Representation

We define the relation in a sentence as the context between two entities (E1, E2) which can be represented as $S = E_1\ W_1\ W_2 \ldots W_n\ E_2$, where W_i represents the ith word between the entities. For instance, let us consider the following sentence taken from CHEMPROT training set:

- **S:** In this report, we show that the hypolipidemic agent atorvastatin is a competitive inhibitor of porcine DPP-IV in vitro, with K(i)=57.8+/-2.3 microM.
- **CR:** [E1 (CHEMICAL)] is a competitive inhibitor of [E2 (GENE-Y)]

Each sentence (S) in the data set is converted into the form of a candidate relation (CR). We then use two different approaches to represent the candidate relations, word embeddings and distance embeddings.

a) Word Embeddings: Distributed word representation (word embeddings) models have gained immense attention as the information load provided a powerful source for unsupervised learning. Word embeddings bring out the semantic aspect of the words by considering the context they usually appear. In this study, we used Gensim [11] implementation of the Word2Vec [12] algorithm that learns fixed-sized continious vectors for each word in the given corpus. We trained the model by using a subset of the Open Access Subset of PubMed Central (http://www.ncbi.nlm.nih.gov/pmc/) dataset of ~37K articles. The size of the output word vectors was set to 200 and the Skip-Gram approach was employed.

Each candidate relation in a sentence was represented as V x d_w matrix where V is equal to the number of words in the candidate relation (vocabulary) and d_w is the dimensionality of the word embedding (i.e. d_w=200). V was set to the size of the longest sentence and zero padding was used.

b) Distance Embeddings: Word Position embeddings (WPE) or distance embeddings encode the relative distance between W_i (i^{th} word) and the two entities (E_1 and E_2). For instance the relative distances of the word "*inhibitor*" in the example sentence (CR) to entities E1 "*atorvastatin*" and E2 "*porcine DPP-IV*" are -4 and 2, respectively.

For each unique distance, a real-valued d_d sized embedding was randomly initialized (d_d = 50).

C. Convolutional Neural Networks (CNNs)

After creating the representation of the relation, we employed 1D convolutions followed by max-pooling operation in order to learn more enriched features from the input. Finally, the model was completed with Fully-Connected (FC) layer. Figure 1 illustrates the CNN-model that we built to predict groups of CPRs.

The convolution layer contains filters (feature maps) that are important in detecting hidden motifs in a sequence. Then pooling layer aggregates the features extracted from the convolutions and reduces the size of the representation and the parameters.

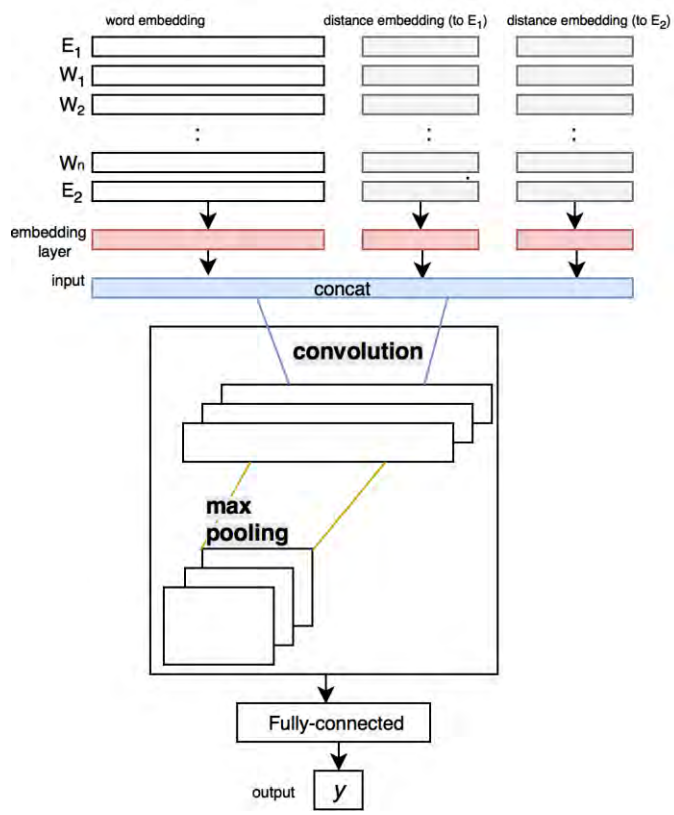


Fig. 1. CNN-based relation classification model

We utilized Keras [13] environment with Tensorflow [14] background to develop the proposed model.

III. RESULTS

The evaluation of the system is reported in precision, recall and F-measure metrics. The performance of our CNN-based system in ChemProt Task is shown in Table II.

This work is funded by Bogazici University Research Fund (BAP) Grant Number 12304.

TABLE II. THE PERFORMANCE OF THE SYSTEM

	Precision	Recall	F-measure
CNN (test)	0.60	0.11	0.18
CNN (dev)	0.99	0.52	0.68

We used CNN architecture depicted in Figure 1 with an extra layer of dropout (0.5) in order to prevent over-fitting. We employed total 100 filters with the length of 3. Softmax was used as activation function and Adam was employed as an optimizer. In the Fully-connected (FC) layer we used 100 hidden nodes. The learning was completed with an epoch of 100.

The proposed system performed significantly better on the development set than the test set in terms of F-measure.

IV. CONCLUSION

In this study, we presented a CNN-based model to extract and classify chemical-protein interactions from the biomedical text using the manually annotated CHEMPROT corpus. We used word-embeddings and distance-embeddings as the features of a candidate relation.

The proposed system achieved F-measure performance of 0.68 on the development set, but performed poorly on the test set with F-measure of 0.18. The system can be further improved to include dependency-based features, attention layers with CNN which are reported to be good at giving higher weights to the important features.

ACKNOWLEDGMENT

TUBITAK-BIDEB 2111 Scholarship Programme is gratefully acknowledged.

REFERENCES

1. Krallinger, Martin, et al. "Overview of the protein-protein interaction annotation extraction task of BioCreative II." Genome biology 9.2 (2008): S4.

2. Abacha, Asma Ben, et al. "Text mining for pharmacovigilance: Using machine learning for drug name recognition and drug–drug interaction extraction and classification." Journal of biomedical informatics 58 (2015):122-132.

3. Liu, Shengyu, et al. "Drug-drug interaction extraction via convolutional neural networks." Computational and mathematical methods in medicine 2016 (2016).

4. Pletscher-Frankild, Sune, et al. "DISEASES: Text mining and data integration of disease–gene associations." Methods 74 (2015): 83-89.

5. Chang, D. T. H., Ke, C. H., Lin, J. H., & Chiang, J. H. (2012). AutoBind: automatic extraction of protein–ligand-binding affinity data from biological literature. Bioinformatics, 28(16), 2162-2168.

6. Chan, W. K., Zhang, H., Yang, J., Brender, J. R., Hur, J., Özgür, A., & Zhang, Y. (2015). GLASS: a comprehensive database for experimentally validated GPCR-ligand associations. Bioinformatics, btv302.

7. Zeng, Daojian, et al. "Relation Classification via Convolutional Deep Neural Network." COLING. 2014.

8. Li, Fei, et al. "A neural joint model for entity and relation extraction from biomedical text." BMC bioinformatics 18.1 (2017): 198.

9.Santos, Cicero Nogueira dos, Bing Xiang, and Bowen Zhou. "Classifying relations by ranking with convolutional neural networks." arXiv preprint arXiv:1504.06580 (2015).

10. Nguyen, Thien Huu, and Ralph Grishman. "Relation Extraction: Perspective from Convolutional Neural Networks." VS@ HLT-NAACL. 2015.

11. Řehůřek, R., and P. Sojka. "Gensim–Python Framework for Vector Space Mo delling." NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic (2011).

12. Mikolov, Tomas, et al. "Distributed representations of words and phrases and their compositionality." Advances in neural information processing systems. 2013.

13. Chollet, François. "Keras." (2015).

14. Abadi, Martín, et al. "Tensorflow: Large-scale machine learning on heterogeneous distributed systems." arXiv preprint arXiv:1603.04467 (2016).

Predicting Chemical Protein Relations with Biaffine Relation Attention Networks

Patrick Verga and Andrew McCallum
College of Information and Computer Sciences, University of Massachusetts Amherst

***Abstract*— Predicting relationships between biological entities is important for drug discovery and precision medicine. The Biocreative VI Task 5 focuses on mining chemical-protein interactions from text. Our submission uses a biaffine relation attention network to encode the full paper abstract and predict relationships between all mention pairs simultaneously. We use no patterns, rules, hard written features, or external resources. Despite this, our best run achieves an test set Micro F1 score of 45.82.**

Keywords—relation extraction, neural networks, chemical protein interactions

I. Introduction

Knowledge bases containing relationships between entieis are powerful tools for downstream tasks such as question answering, query understanding, and exploratory research such as drug discovery and precision medicine. Because these knowledge bases are highly incomplete, methods for knowledge base completion have been developed. These can broadly be broken down into link predition – inferring missing links using existing graph properties, and relation extraction – mining new entities and relationships from text.

Extracting relations between entities is one of the core problems in information extraction. Initial methods relied on hand written patterns and bootstrapping methods. Later rich hand crafted features were fed into machine learning classifiers such as support vector machines. More recently, neural networks have become the state of the art, primarily gated recurrant neural networks and convolutional neural networks.

Previous neural models for relation extraction have formed predictions on a single mention pair at a time constrained to a single sentence. Our model instead produces all predictions for all mention pairs simultaneously by encoding the entire abstract. The biaffine relation attention network (BRAN) encodes the full paper abstract using the Transformer attention-based architecture [2] and then applies a bi-affine operation between all mention pairs with respect to the set of query relations [1].

Our experiments on the Biocreative VI Task 5: Text mining chemical-protein interactions (CHEMPROT) dataset show our models strong performance. Our model encoding the entire abstract outperforms an equivalent sentence level classifier by incorporating a broader context to make its predictions. We improve our performance further by ensembling many versions of our model trained with different random seeds.

II. Model

The BRAN model was first proposed in [1] and was shown to have state of the art performance on the Biocreative V Chemical Disease Relation dataset. The model does not use any handcrafted features or rules. Even tokenization is performed using corpus statistics.

A. Inputs

We tokenize our data into byte-pair encodings using a budget of 7500 [3]. The full abstract is converted to a sequence of token embeddings of dimensions 64. Words are randomly replaced with a special UNK token with probability .15 and we apply dropout to the embeddings with probability .15.

B. Transformer

These token embeddings are then contextually encoded using the Transformer architecture with 2 block repeats and internal dimensions of size 64. The feed-forward component consists of 3 convolutional layers with kernel width 1, 5, and 1 and dimension 256. Dropout is applied to the internal layers with probability .15.

C. Biaffine pairwise scores

For each pair of tokens in our abstract we compute a biaffine operation using a 64 by 64 dimensional matrix per relation type.

D. Training

We train our model using cross entropy over the training set using the Adam optimizer with learning rate .0005, epsilon 1e-4, beta1 .9 and beta2 .9 and a batch size of 8. We additionally apply add gradient noise with standard deviation .1 [4]. Additionally, we use the final output representations of the Transformer to predict named entity labels using BIO encoding. We perform early stopping on the development set optimized for Micro F1.

III. Results

Model	*F1*
Sentence Ensemble	44.6
Abstract Ensemble	46.9

Fig. 1. F1 scores on the development set. Our model trained on full asbtracts outperforms our model trained on just single sentences. Example of a figure caption.

Model	*F1*
Abstract	41.2
Abstract Ensemble	46.9
Abstract+Sentence Ensemble	49.8

Fig. 2. F1 scores on the development set. Ensemble models outperform the single model scores. Combining models trained on the full abstract and those trained on the single sentences improves performance further.

Model	*Precision*	*Recall*	*F1*
Abstract+Sentence Ensemble	47.18	44.53	45.82

Fig. 3. Test set scores.

Figure 1 shows various versions of our models scores on the development set. We tune each of the per-relation decision thresholds separately. We apply a post-processing step to our full abstract models to remove predictions that cross the sentence boundary because the CHEMPROT dataset only considers within sentence relationships. Models denoted with ensemble average the prediction probabilities of 20 models trained with the same hyper parameters but different random seeds.

In Figure 1 we see that our model that is trained on the full abstract outperforms the model trained on single sentences. Figure 2 shows that ensembling models outperforms single models, and ensembling the sentence level and abstract level models improve performance further. Figure 3 shows the results of our best run (run 4) which ensembles the full abstract and sentence models.

IV. Conclusion

We presented our submission to the Biocreative VI Task 5 Chemical-Protein relation text mining shared task. Our biaffine relation attention model effectively and efficiently extracted chemical protein relationships despite using no hand crafted features or external resources.

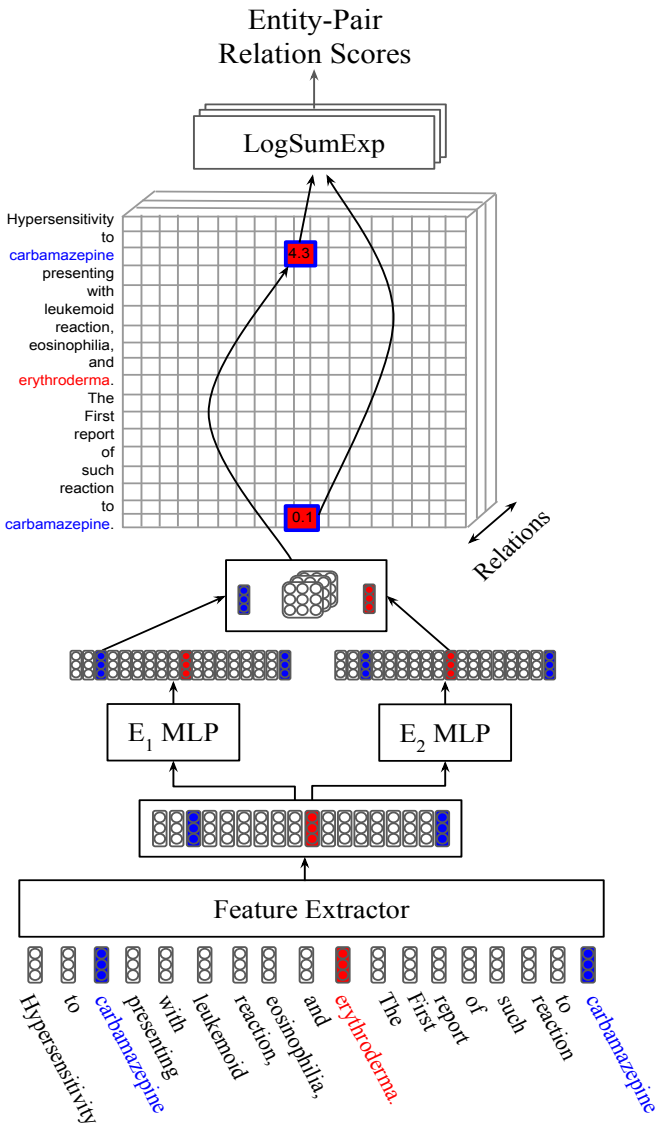


Fig. 4. The relation extraction architecture. Inputs are initially contextually encoded using the transformer. Each transformed token is then passed through an $Entity_1$ and $Entity_2$ MLP to produce two separate versions of each token able to act as either the head or tail of the relationship. A bi-affine operation is then performed between each $Entity_1$ and $Entity_2$ token with respect to each relations embedding matrix producing a token $\times$ token $\times$ relation pairwise affinity tensor. Finally, the scores for cells corresponding to the same entity pair are pooled with a separate LogSumExp operation for each relation to get a final score. The colored tokens are meant only to illustrate calculating the score for a given pair of entities. The model is only given entity information when gathering scores to pool from the affinity matrix.

REFERENCES

1. Verga, P., Strubell, E., Shai, O., McCallum, A.. (2017) Attending to All Mentions for Full Abstract Biological Relation Extraction. *arXiv preprint*.

2. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L. and Polosukhin, I., 2017. Attention Is All You Need. arXiv preprint arXiv:1706.03762.

3. Neelakantan, A., Vilnis, L., Le, Q.V., Sutskever, I., Kaiser, L., Kurach, K. and Martens, J., 2015. Adding gradient noise improves learning for very deep networks. arXiv preprint arXiv:1511.06807.

4. Sennrich, R., Haddow, B., and Birch, A.. (2015). Neural machine translation of rare words with subword units. *arXiv preprint*

Chemical-Gene Relation Extraction Using Recursive Neural Network

BioCreative Challenge VI track 5 submission

Sangrak Lim, Jaewoo Kang
Affiliation: Department of Computer Science and Engineering, Korea University, Seoul, Korea

***Abstract*— In this paper, we describe our system for the CHEMPROT task of BioCreative VI challenge. Extracting relations between chemical compounds and genes from biomedical literature is an important element to the biomedical application such as Precision Medicine (PM). The CHEMPROT task of BioCreative VI aims to promote the development of the text mining systems that can be used to find relationships between chemical compounds and genes automatically. We use a Recursive Neural Network to improve the performance of relation extraction. Our model uses a position feature, a subtree containment feature, and an ensemble method to improve the performance of CHEMPROT relation extraction. Our system scored 58.53% (F-score) at the CHEMPROT task of BioCreative VI challenge test set.**

Keywords—relation extraction; recursive neural network; text mining

I. INTRODUCTION

There is an increasing interest to find relationships between biological and chemical entities within natural language texts and store the relationship information in the form of structured databases. In the biomedical field, many new papers are published on the relationship between chemicals and genes. Since this relationship information is dispersed in each paper, it is necessary to extract and gather the relationship first that might benefit the biomedical application such as Precision Medicine (PM). PM refers to determining prevention and treatment strategies according to the individual variability to provide a more specific and effective treatment (1). Gonzalez et al. (2) mentioned that manual curation of relation extraction is still the current standard because the value of the text mining applications still have room for improvement. However, it is an laborious task to extract the relationships information by hand. To solve this problem, the CHEMPROT track in BioCreative VI is intended to facilitate the development of systems that automatically extract relationships from text in natural language and classifies them into several types that are important to biology.

The CHEMPROT track in BioCreative VI organizers manually annotated chemical-gene entity relations in abstracts and divide the relation types into 10 classes. Although all the classes are important in the biochemical and pharmacological perspective, only five groups are used for the evaluation. Table 1. shows the five groups and one example sentence for each group. The data given in the challenge consists of abstracts, entities, and relations between two entities. Each entity has position information in the abstract and an entity with the same name appears multiple times. For the example in the Table 1., we found that almost all the relations take place between the two entities in a same sentence. In the pre-processing step, we split the abstract into sentences and consider a sentence with two entities to have a candidate relationship. Details are covered in the section Ⅱ.

The CPR:3 is related with the upregulation and usually associated with words like “activate”, “promote”, and “increase activity of”. The CPR:4 is related with the downregulation and usually associated with words like “inhibitor”, “block”, and “decrease activity of”. The CPR:5 and CPR:6 is related with agonist and antagonist, respectively. These four groups all have distinctive features. However, when multiple entities co-occur in a sentence, it is difficult to determine if a relationship holds between the two target entities of interest. The CPR:9 is related with substrate. Unlike the above four groups, the CPR:9 do not have noticeable features, and thus the relation is difficult to extract.

We build a relation extraction system for the CHEMPROT track in BioCreative VI using the Recursive Neural Network (Recursive NN) based approach. Our Recursive NN model uses syntactical features of each node in a parse tree. Socher et al. (3) claimed that grammatical structure of natural language sentences is recursive. We believe that Recursive NN approach is effective for relation extraction task.

II. METHODS

Our system architecture is presented in the Fig 1.

A. Preprocessing

Preprocessing involves sentence splitting, and anonymizing target entities and chemical compounds. The abstract data consists of several sentences. We found that almost all the relations take place between the two target entities in a same sentence; therefore, we split the abstract into sentences and consider a sentence with two entities to have a candidate

Table 1. Five groups of CHEMPROT relations to be used for evaluation

Group	*CHEMPROT relations*	*Sentence Example*
CPR:3	UPREGULATOR|ACTIVATOR| INDIRECT_UPREGULATOR	<BC6ENT1>Amitriptyline</BC6ENT1>, but not any other tricyclic or selective serotonin reuptake inhibitor antidepressants, promotes <BC6ENT2>TrkA</BC6ENT2> autophosphorylation in primary neurons and induces neurite outgrowth in PC12 cells.
CPR:4	DOWNREGULATOR|INHIBITOR| INDIRECT_DOWNREGULATOR	Ginseng total saponins, <BC6ENT1>ginsenosides Rb2, Rg1 and Rd</BC6ENT1> administered intraperitoneally attenuated the immobilization stress-induced increase in plasma <BC6ENT2>IL-6</BC6ENT2> level.
CPR:5	AGONIST|AGONIST-ACTIVATOR| AGONIST-INHIBITOR	At 10(-6)M in transcription assays, none of these compounds showed progestin agonist activity, whereas <BC6ENT1>mifepristone</BC6ENT1> and its monodemethylated metabolite manifested slight <BC6ENT2>glucocorticoid</BC6ENT2> agonist activity.
CPR:6	ANTAGONIST	In another experiment, <BC6ENT1>cyanopindolol</BC6ENT1>, an antagonist of the <BC6ENT2>serotonin terminal autoreceptor</BC6ENT2>, also prolonged the clearance of 5-HT from the CA3 region.
CPR:9	SUBSTRATE|PRODUCT_OF| SUBSTRATE_PRODUCT_OF	Leukotriene A(4) hydrolase (<BC6ENT1>LTA(4)H</BC6ENT1>) is a cystolic enzyme that stereospecifically catalyzes the transformation of <BC6ENT2>LTA(4)</BC6ENT2> to LTB(4).

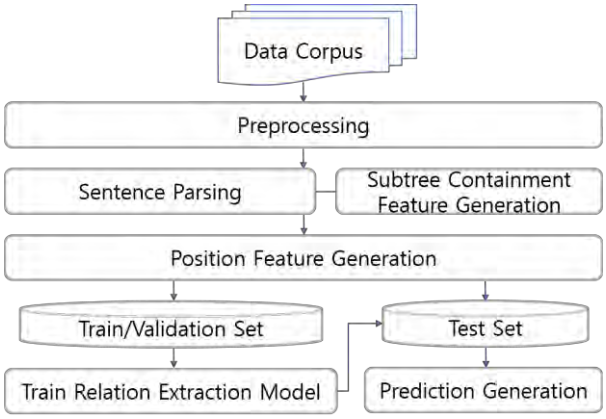


Figure 1. Overall System Architecture

relationship. For the training data, we assign group labels to the candidate relationship.

The biochemical entities usually have long and complex names. To reduce the complexity, we replace the target entities of interest with "BC6ENT1" and "BC6ENT2" in the order in which they appear. For the other entity mentions in a sentence, we apply the ChemDataExtractor [1] to find the chemical entities and replace them with "CHEM" (4).

B. Parsing Sentences

Our Recursive NN model in Natural Language Processing (NLP) uses syntactical features of each node in a parse tree. We use the Stanford NLP library to transform a sentence into a parse tree. After the parsing process, we use the "binarizer" provided by the Stanford Parser to convert the parse tree into a binary tree.

C. Subtree Containment Feature Generation

We calculate the subtree containment feature during the parsing stage. The subtree containment feature indicates that certain subtree contains an important entity. When one of the target entities exist in the leaves of the current node, the subtree containment feature is given a value of one; otherwise, it is given a value of zero. This feature is later converted into a vector with a size 10. If the value is one, every element of a vector is one; otherwise, every element in a vector is zero.

D. Position Feature Generation

Position feature embedding represents the relative distance of two target entities from each word position in a sentence (5). Every word in a sentence has two relative distances, $[d_1, d_2]$, where d_i is the relative distance to i_{th} target entity from the current word. In the training phase, each relative distance is converted into a vector with a size of 10. Since there are two distances, the total vector size of the position feature embedding is 20. Figure 2. shows the vector representation based on the relative distances. Note that when the distance difference is 5 or less, the vector is assigned to each difference value. If the distance is greater than 5, the same vector is given in units of 5. We skip the columns ranged from -2 to -∞ of the relative distance due to space limitation.

E. Word Embedding

Word embedding is a set of low-dimensional vectors that are trained by an unsupervised language model. Word embedding combined with a neural network is a widely method to improve NLP performance (8,9). We used the PubMed-and-PMC-w2v word embedding, which is obtained from published materials [2] (10). The word embedding is initialized with Word2Vec using genism (11). The dimension size of the word embedding is 200.

F. Recursive Neural Network with TreeLSTM

The Long Short-Term Memory (LSTM) architecture is a popular variation of the recurrent neural network (6). The general LSTM is used for sequential data, such as sentences. We implemented tree-LSTM to apply the LSTM architecture for our tree-structured data (7). A node in a tree-LSTM receive input from multiple child nodes and update the hidden state of current node using the input.

[1] http://chemdataextractor.org/download

[2] http://evexdb.org/pmresources/vec-space-models/

relative distance	-1	0	1	2	3	4	5	6~10	11~15	16~20	21~∞
	0	0	1	1	1	1	1	1	1	1	1
	0	0	0	0	0	0	0	0	0	0	1
	0	0	0	0	0	0	0	0	0	1	1
	0	0	0	0	0	0	0	0	1	1	1
	0	0	0	0	0	0	0	1	1	1	1
	0	0	0	0	0	0	1	1	1	1	1
	0	0	0	0	0	1	1	1	1	1	1
	0	0	0	0	1	1	1	1	1	1	1
	0	0	0	1	1	1	1	1	1	1	1
	1	0	1	1	1	1	1	1	1	1	1

Figure 2. Vector representation according to the distance between one of the target entities and a current word.

Table 2. Search process to find the best hyperparameter

Parameter	Test Range	Test Unit	Selected
Hidden Unit Size	64 - 256	64	256
Subtree Containment Size	1 - 10	2	10
Batch Size	64 - 256	64	256
Learning Rate	0.0005 – 0.01	0.0005	0.001
Keep Probability	0.5 – 0.9	0.1	0.5

After receiving a parse tree to train our model, we look up the pre-trained word embedding to assign real-valued vectors to each word. If a node is not a leaf, the word vector is randomly initialized. Our model is based on the recursive NN architecture of the child sum tree-LSTM model (7).

Let x_j denote the concatenation result of the vector representation of a word with feature vectors. For any node j, we have two forget gates for each child and write the sub-node expression of the forget gates for k-th child as f_{jk}. The $B(j)$ is the set of values (including h_k and c_k) from children of node j, since we use a binary tree, the size of $B(j)$ is 2. i, f, o, c, h are the input gate, forget gate, output gate, memory cell, and the hidden state respectively. u is a temporary value that could be added to the memory cell state. $drop(x)$ is a recurrent dropout function (12). The $mask$ is a sampled vector from the Bernoulli distribution with success probability $keep_p$. Our tree-LSTM equations are described below.

$$\tilde{h_j} = \sum_{k \in B(j)} h_k, \tag{1}$$

$$i_j = \sigma(W^i[x_j, \tilde{h_j}] + b^i), \tag{2}$$

$$f_{jk} = \sigma(W^f[x_j, h_k] + b^f), \tag{3}$$

$$o_j = \sigma(W^o[x_j, \tilde{h_j}] + b^o), \tag{4}$$

$$u_j = tanh(W^u[x_j, \tilde{h_j}] + b^u), \tag{5}$$

$$c_j = i_j \odot drop(u_j) + \sum_{k \in B(j)} f_{jk} \odot c_k, \tag{6}$$

$$h_j = o_j \odot tanh(c_j) \tag{7}$$

$$drop(x) = \begin{cases} mask * x, & if \quad train \quad phase, \\ x, & otherwise \end{cases} \tag{8}$$

Equation (9,10) is a fully-connected layer we use as the output layer. The fully-connected layer output size is the same as the number of classes (six, one for false class, five for the group classification). At each node j, we choose the predicted label $\hat{y_j}$ for a given output. However, since the predicted value of the internal nodes in the tree is not important, we take only the predicted values extracted from the root node of the entire sentence when the final score is calculated. We use the softmax cross-entropy classifier to calculate the cost function. m is the total number of items in the training set.

$$\hat{p}(y|x_j) = W^{(fc)}h_j + b^{(fc)} \tag{9}$$

$$\hat{y} = \arg\max_y \hat{p}(y|x_j) \tag{10}$$

$$J(\theta) = -\frac{1}{m}\sum_k^m y^k log(softmax(\hat{p}(y^k|x^k))) \tag{11}$$

We use the Adam optimizer for gradient descent optimization. An input vector of a node in a tree uses the subtree containment feature vector, the position feature vector, and the vector representation of a word in a sentence. The size of the whole input vector x_j is 10 + 20 + 200.

G. Regularization

The original tree-LSTM model (7) used l2 regularization. The tree-LSTM model was implemented with the Tensorflow fold library (13) using recurrent dropout (12) instead of the l2 regularization. We found that recurrent dropout is effective.

H. Ensemble method

The random weight initialization changes the result of neural networks considerably (14). Since the CHEMPROT task is challenging, it is difficult to reproduce the exact same result for the single model, and we resolve this problem to some extent using the ensemble model. We sum the output probabilities (logits) of ensemble members, which are from the same repeated experiment for the evaluation. Our model using ensemble method utilizes four (for the first run) and six (for the second run) ensemble members.

III. Results

A. Experimental Setting

We use Tensorflow to implement our model (15). Most deep learning libraries such as Tensorflow assume machine learning models are static, which makes it difficult to use them with dynamic structured models (e.g., Recursive NN). We implement our Recursive NN model with Tensorflow Fold (13). The Tensorflow Fold is specifically designed to deal with dynamic structured problem.

B. Hyperparameter

We found the optimal parameters by moving one parameter within the specified test range by a specified test unit while other parameters were fixed. Table 2. illustrates the hyperparameter search process. Because the CHEMPROT task support the development set, we searched the hyperparameter on the development set, while the model was trained with the training set.

Table 3. The statistics of the BioCreative VI CHEMPROT corpus after preprocessing

	Abstracts	Positive	Negative	Total	Ratio
Train Set	1,020	4,157	12,807	16,964	1:3.08
Development Set	612	2,416	8,198	10,614	1:3.39
Test Set	3,399	-	-	58,523	-

Table 4. CHEMPROT results of our Recursive NN system on the test set.

Run	Precision	Recall	F-Score
First	0.6760	0.5159	0.5852
Second	0.6704	0.5194	0.5853

C. Data corpus

After we preprocess the given data, the number of negative instances is more than three times larger than the number of positives. Table 3. shows the statistics of the preprocessed corpus. We combined the training and development set as a training set for the final model.

D. Performance

The results of our system on the test dataset are shown on Table 4. We submitted two runs for different number of ensemble members. Our model using ensemble method utilizes four (for the first run) and six (for the second run) ensemble members. The best F1-score of our system is the 58.53% in the second run.

IV. CONCLUSION

We implemented the tree-LSTM architecture to understand the natural language sentences. Since we do not know the labels of the test data, we could not proceed the ablation study of additional features on the test set. Still, when we checked our model in validation data, using additional features showed better performance.

The source of our Recursive NN project is available at : https://github.com/arwhirang/recursive_chemprot

REFERENCES

1. Collins, F. S., & Varmus, H. (2015). A New Initiative on Precision Medicine. The New England Journal of Medicine, 372(9), 793–795.

2. Gonzalez, G. H., Tahsin, T., Goodale, B. C., Greene, A. C., & Greene, C. S. (2016). Recent Advances and Emerging Applications in Text and Data Mining for Biomedical Discovery. Briefings in Bioinformatics, 17(1), 33–42.

3. Socher R., Lin C. C., Manning C., and Ng A. Y. (2011). Parsing natural scenes and natural language with recursive neural networks. In Proceedings of the 28th international conference on machine learning (ICML-11);pages 129–136

4. Swain, M. C., & Cole, J. M. (2016). ChemDataExtractor: A Toolkit for Automated Extraction of Chemical Information from the Scientific Literature. Journal of chemical information and modeling, 56(10), 1894-1904.

5. Zeng, D., Liu, K., Lai, S., Zhou, G., & Zhao, J. (2014). Relation Classification via Convolutional Deep Neural Network. In International Conference on Computational Linguistics (COLING) (pp. 2335-2344).

6. Hochreiter S. and Schmidhuber J. (1997). Long short-term memory. Neural computation. 9(8),1735–1780.

7. Tai K. S., Socher R., and Manning C. D. (2015) Improved semantic representations from tree-structured long short-term memory networks. arXiv preprint arXiv:1503.00075

8. Bengio Y., Ducharme R., Vincent P., and Jauvin C. (2003). A neural probabilistic language model. journal of machine learning research, 1137–1155.

9. Mikolov T., Chen K., Corrado G., and Dean J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

10. Van Landeghem S., Björne J., Wei C. H., Hakala K., Pyysalo S., Ananiadou S.,Kao H. Y., Lu Z., Salakoski T., Van de Peer Y., et al. (2013). Large-scale event extraction from literature with multi-level gene normalization. PloS one, 8(4), e55814.

11. Řehůřek R. and Sojka P. (2010). Software Framework for Topic Modelling with Large Corpora. In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, pages 45–50

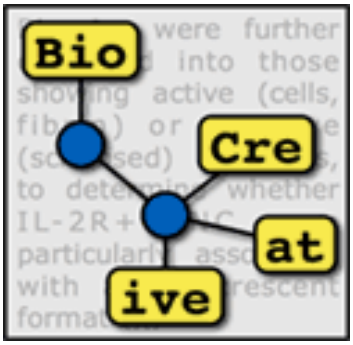
12. Semeniuta S., Severyn A., and Barth E. (2016). Recurrent dropout without memory loss. arXiv preprint arXiv:1603.05118.

13. Looks M., Herreshoff M., Hutchins D., and Norvig P. (2017) Deep learning with dynamic computation graphs. arXiv preprint arXiv:1702.02181

14. Kadlec R., Schmid M., Bajgar O., and Kleindienst J. (2016). Text understanding with the attention sum reader network. arXiv preprint arXiv:1603.01547.

15. Abadi M., Agarwal A., Barham P., Brevdo E., Chen Z., Citro C., Corrado G. S.,Davis A., Dean J., Devin M, et al. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint. arXiv:1603.04467.

General Session



Efficient and Accurate Entity Recognition for Biomedical Text

Fabio Rinaldi[1], Lenz Furrer[1], Marco Basaldella[2]
1 University of Zurich, 2 Università degli Studi di Udine

Abstract- **This short paper briefly presents an efficient implementation of a named entity recognition system for biomedical entities, which is also available as a web service. The approach is based on a dictionary-based entity recognizer combined with a machine-learning classifier which acts as a filter. We evaluated the efficiency of the ap-proach through participation in the TIPS challenge (BioCreative V.5), where it obtained the best re-sults among participating systems. We separately evaluated the quality of entity recognition and link-ing, using a manually annotated corpus as a reference (CRAFT), where we obtained state-of-the-art results.**

Keywords- named entity recognition; text mining; machine learning; natural language processing.

I. INTRODUCTION

Named entity recognition is most often tackled with knowledge-based approaches (using dictionaries) or example-based approaches (machine learning). Currently the best results are obtained using supervised machine-learning based systems. For extracting chemical names, [9] describes how two CRF classifiers are trained on a corpus of journal abstracts, using different features and model parameters. The approach in [10] also tackles chemical name extraction with CRF, partly using the same software basis as the previous one. For tagging gene names, [14] describes another supervised sequence-labeling approach, using a CRF classifier.

There is growing interest in hybrid systems combining machine learning and dictionary approaches such as the one described in [1], which obtains interesting performance on chemical entity recognition in patent texts.

In the field of entity linking, dictionary-based methods are predominant, since the prediction of arbitrary identi-fiers cannot be modeled in a generalized way. In [6], the authors explore ways to improve established information retrieval techniques for matching protein names and other biochemical entities against ontological resources. The TaggerOne system [8] uses a joint model for tackling NER and linking at the same time – yet another example of a hybrid system that combines machine learning and dictionaries.

II. DATA

The Colorado Richly Annotated Full Text (CRAFT) cor-pus [2, 16] has been built specifically for evaluating these kinds of systems. It consists of 67 full-text articles that have been manually annotated with respect to chemicals, genes, proteins, cell types, cellular components, biological processes, molecular functions, organisms, and biological sequences. In total, the available articles are annotated with over 100,000 concepts.

For our experiments, we used all terminology resources that were distributed with the corpus (which means all an-notated entities, except those grounded using Entrez Gene). We regarded species and higher taxonomic ranks (genus, or-der, phylum etc.) from both cellular organisms and viruses as a common entity type "organism". Also, we combined the two non-physical entity types (biological processes and molecular functions) into a single class.

III. METHODS

The OntoGene group has developed an approach for biomedical entity recognition based on dictionary lookup and flexible matching. Their approach has been used in several competitive evaluations of biomedical text mining technologies, often obtaining top-ranked results [12, 13, 11]. Recently, the core parts of the pipeline have been imple-mented in a more efficient framework using Python [4] and are now developed under the name OGER (OntoGene's En-tity Recognizer). These improvements showed to be effec-tive in the BioCreative V.5 shared task [7]: in the techni-cal interoperability and performance of annotation servers (TIPS) task, our system achieved best results in four out of six evaluation metrics [5].

OGER offers a flexible web API for performing dictionary-based NER. It accepts a range of input formats and provides the annotated terms along with identifiers in various output formats. We run an instance of OGER as a permanent web service which is accessible through an API and a web user interface.[1]

For the experiments with the CRAFT corpus, we used the ontologies on which the original annotation was based. We use those resources to compile a non-hierarchical dictionary with 1.26 million terms pointing to 864,000 concept identifiers.
The input documents were tokenized with a simple method based on character class, which collapsed spelling variants such as "SRC 1", "SRC-1", and "SRC1" to a common form.

[1] `https://pub.cl.uzh.ch/projects/ontogene/oger/`

TABLE 1: Performance of our system in entity recognition (top) and entity linking (a.k.a. concept recognition, bottom), compared to the best results reported in [15].

System	Precision	Recall	F1
OGER	0.59	**0.66**	0.62
OGER+NN	**0.86**	0.60	**0.70**
OGER	0.32	**0.52**	0.40
OGER+NN	**0.51**	0.49	**0.50**
cTakes Dict. Lookup	**0.51**	0.43	0.47

All tokens were then converted to low-ercase and stemmed, except for acronyms that collide with a word from general language (e.g. "WAS"). As a further normalization step, Greek letters were expanded to their letter name in Latin spelling, e.g. "α" → "alpha".

In order to improve the system's accuracy, we added a machine-learning filter to remove spurious matches. We used an approach based on neural networks (NN), as they were the best performing algorithm in our previous exper-iments described in [3]. Training is performed using 10-fold cross validation on 47 articles; the evaluation is thus performed on 20 documents only. The features used are mostly shape-based (character count, capitalization), but some include linguistic information (POS, stem) or domain knowledge (frequent pre-/suffixes).

IV. RESULTS

We examined our system in two separate evaluations. We first considered the performance of NER proper, i.e. we re-garded only offset spans and the (coarse) entity type of each annotation produced by each system, ignoring con-cept identifiers. We then evaluated the correctness of the selected concept identifiers. To this end, we augmented the ML-based output with concept identifiers taken from the dictionary-based pre-annotations, which enabled us to draw a fair comparison to previous work in entity linking on the CRAFT corpus.

A. Named Entity Recognition We have compiled very de-tailed results for different configurations and for each en-tity category, however the brevity of this paper allows us to present only aggregated results. The OGER pipeline alone (without filtering) delivers an overall 66% recall score with a precision of 59% over all the entity types consid-ered. Adding the NN-based filtering module, recall drops to 60%, with an increase in precision to 86%, leading to a very competitive F-score of 70%.

B. Concept Recognition We chose a simple strategy to reintroduce the concept identifiers provided by OGER into the output of the ML systems, based on the intersection of the original annotations from OGER's output (which in-clude identifiers) and the annotations left after applying by the NN-based filter. We did not resolve ambiguous an-notations; instead, multiple identifiers could be returned for the same span. While having no disambiguation at all is arguably a deficiency for an entity linking system, it is not imperative that each and every ambiguity is reduced to a single choice. This is particularly true when evaluating against CRAFT, which contains a number of reference an-notations with multiple concept identifiers. For example, in PMID: 16504143, PMCID: 1420314, the term "fish" (oc-curring in the last paragraph of the Discussion section) is assigned six different taxonomic ranks.

This simple strategy allows the system to reach a pre-cision of 51% with a recall of 49% in concept recognition. Compared with the results of several previous systems re-ported in [15], who carried out a series of experiments using the same dataset, our results are already state-of-the-art.

Please note that the results reported by [15] are not per-fectly comparable to the ones we obtained, since the former were tested on the whole CRAFT corpus, while our approach was evaluated on 20 documents only (since we used the remaining documents to train our system) Still, the comparison shows that even a relatively simple approach is sufficient to transform our NER pipeline into an entity linking system with reasonable quality. This is particularly true for the OGER-NN configuration, where both precision and recall are as good as or better than the figures for all the reported systems.

V. CONCLUSION AND FUTURE WORK

In this paper, we presented an efficient, high-quality system for biomedical entity recognition and linking (OGER). We evaluated both processing speed and annotation quality in a series of in-domain experiments using the CRAFT corpus. OGER's scalability and efficiency was also demonstrated in the recently held TIPS task of the BioCreative V.5 chal-lenge. For the NER performance, we used a NN classifier, which acted as a postfilter of the dictionary annotations. The combined system achieved competitive results in en-tity recognition and state-of-the-art results in entity linking over the selected evaluation corpus (CRAFT).

Currently we expose via web API only the OGER service (entity recognition and linking) but without disambigua-tion. As a next step in this research activity, we intend to make available a second web service including disam-biguation. At the same time we are performing additional experiments aimed at improving the quality of the disam-biguation step.

REFERENCES

[1] Saber A Akhondi, Ewoud Pons, Zubair Afzal, Herman van Haagen, Benedikt FH Becker, Kristina M Hettne, Erik M van Mulligen, and Jan A Kors. Chemical entity recognition in patents by combining dictionary-based and statistical approaches. *Database*, 2016:baw061, 2016.

[2] Michael Bada, Miriam Eckert, Donald Evans, Kristin Garcia, Krista Shipley, Dmitry Sitnikov, William A Baumgartner, K Bretonnel Cohen, Karin Verspoor, Judith A Blake, et al. Concept annotation in the CRAFT corpus. *BMC bioinformatics*, 13(1):1, 2012.

[3] Marco Basaldella, Lenz Furrer, Nico Colic, Tilia R Ellendorff, Carlo Tasso, and Fabio Rinaldi. Using a hybrid approach for entity recognition in the biomedical domain. In *Proceedings of the 7th International Symposium on Semantic Mining in Biomedicine (SMBM 2016)*, 2016.

[4] Nicola Colic. Dependency parsing for relation extraction in biomedical literature. Master's thesis, University of Zurich, Switzerland, 2016.

[5] Lenz Furrer and Fabio Rinaldi. OGER: OntoGene's entity recogniser in the BeCalm TIPS task. In *Proceedings of the BioCreative V.5 Challenge Evaluation Workshop*, pages 175–182, 2017.

[6] Tudor Groza and Karin Verspoor. Assessing the impact of case sensitivity and term information gain on biomedical concept recognition. *PloS one*, 10(3):e0119091, 2015.

[7] Martin Krallinger, Martin Pérez-Pérez, Gael Pérez-Rodríguez, Aitor Blanco-Míguez, Florentino Fdez-Riverola, Salvador Cappella-Gutierrez, Anália Lourenço, and Alfonso Valencia. The BioCreative V.5/BeCalm evaluation workshop: tasks, organization, sessions and topics. In *Proceedings of the BioCreative V.5 Challenge Evaluation Workshop*, pages 8–10, 2017.

[8] Robert Leaman and Zhiyong Lu. TaggerOne: joint named entity recognition and normalization with semi-Markov Models. *Bioinformatics*, 32(18):2839, 2016.

[9] Robert Leaman, Chih-Hsuan Wei, and Zhiyong Lu. tmChem: a high performance approach for chemical named entity recognition and normalization. *J. Cheminformatics*, 7(S-1):S3, 2015.

[10] Tsendsuren Munkhdalai, Meijing Li, Khuyagbaatar Batsuren, Hyeon Ah Park, Nak Hyeon Choi, and Keun Ho Ryu. Incorporating domain knowledge in chemical and biomedical named entity recognition with word representations. *Journal of Cheminformatics*, 7(1):S9, 2015.

[11] Fabio Rinaldi, Simon Clematide, Hernani Marques, Tilia Ellendorff, Raul Rodriguez-Esteban, and Martin Romacker. OntoGene web services for biomedical text mining. *BMC Bioinformatics*, 15(14), 2014.

[12] Fabio Rinaldi, Thomas Kappeler, Kaarel Kaljurand, Gerold Schneider, Manfred Klenner, Simon Clematide, Michael Hess, Jean-Marc von Allmen, Pierre Parisot, Martin Romacker, and Therese Vachon. OntoGene in BioCreative II. *Genome Biology*, 9(Suppl 2):S13, 2008.

[13] Fabio Rinaldi, Gerold Schneider, Kaarel Kaljurand, Simon Clematide, Therese Vachon, and Martin Romacker. OntoGene in BioCreative II.5. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 7(3):472–480, 2010.

[14] Golnar Sheikhshab, Elizabeth Starks, Aly Karsan, Anoop Sarkar, and Inanc Birol. Graph-based semi-supervised gene mention tagging. In *Proceedings of the 15th Workshop on Biomedical Natural Language Processing*, pages 27–35, Berlin, Germany, 2016. Association for Computational Linguistics.

[15] Eugene Tseytlin, Kevin Mitchell, Elizabeth Legowski, Julia Corrigan, Girish Chavan, and Rebecca S Jacobson. NOBLE – Flexible concept recognition for large-scale biomedical natural language processing. *BMC bioinformatics*, 17(1):1, 2016.

[16] Karin Verspoor, Kevin Bretonnel Cohen, Arrick Lanfranchi, Colin Warner, Helen L. Johnson, Christophe Roeder, Jinho D. Choi, Christopher Funk, Yuriy Malenkiy, Miriam Eckert, Nianwen Xue, William A. Baumgartner, Michael Bada, Martha Palmer, and Lawrence E. Hunter. A corpus of full-text journal articles is a robust evaluation tool for revealing differences in performance of biomedical natural language processing tools. *BMC Bioinformatics*, 13(1):207, 2012.

iTextMine: Integrated Text Mining System for Large-Scale Knowledge Extraction from Literature

Jia Ren[1], Gang Li[2], Cathy H. Wu[1,2]
[1]Center for Bioinformatics and Computational Biology [2]Department of Computer and Information Sciences
University of Delaware, Newark, Delaware, United States of America

***Abstract*—We describe the iTextMine system with an automated workflow to run multiple text-mining tools on large-scale text for knowledge extraction. We employ parallel processing for dockerized text mining tools with a common JSON output format, and implement a text alignment algorithm to align entity offsets in the text for result integration. iTextMine presently consists of four relation extraction tools and has processed all Medline abstracts. The website (http://research.bioinformatics.udel.edu/itextmine) allows users to browse the text evidence and view integrated results for knowledge discovery through a network visualization.**

Keywords—text-mining, relation extraction, text annotation, knowledge integration

I. Introduction

With the rapid growth of biomedical literature, text-mining tools help biologists extract useful information quickly. Most text-mining tools are specialized on specific tasks and may be used to recognize certain types of entities or relations. Thus, there is a need to combine results from different text-mining tools to cover a broad range of bioentities and relations for more comprehensive biological knowledge. Combining articles describing multiple relation types, one may identify cross-talk among different entities and relations extracted by different tools.

However, there are challenges in integrating different tools for large-scale processing and knowledge integration: (i) text-mining tools have different run-time dependencies and it is cumbersome to maintain them in the parallel execution engine. Meanwhile we need to make sure each tool can be run in parallel, e.g., two parallel processes will not write to the same file to avoid conflict; (ii) each tool has its own output format to describe the extracted information, and it is hard to store the result with the same database schema; (iii) some text-mining tools modify original text, and the text offset of entity and relation cannot be matched to the original text, making it impossible to directly compare the results from different tools.

Here we describe iTextMine to address the tool integration challenges. The system has been used for processing the entire set of Medline abstracts, combining and disseminating text-mining results from multiple tools developed in our group. Biologists can use this system to perform knowledge discovery via an interactive interface. The system will be run periodically to update the database when new literature is released.

II. Method

A. System Overview

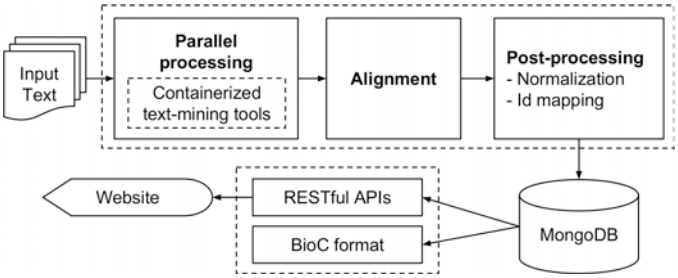


Fig. 1. iTextMine system with an automated workflow to integrate text mining tools and relation extraction results from large-scale text processing

The iTextMine workflow has four major steps (Fig. 1). We first prepare input literature by downloading Medline abstracts from PubMed website. The text is indexed by Lucene[1] on our local machine. Each tool uses a tool-specific entity/trigger word-based query to identify positive abstracts with potential entities or relations of interest. Then a parallel execution engine is set up to run dockerized text-mining tools in parallel. Before importing the text-mining results into the database, text alignment is performed to adjust entity offset. Additional post-processing tasks can be performed afterward, such as entity normalization and ID mapping. The post-processed data is then stored directly into MongoDB[2]. Finally, web services are built to disseminate the results. We created REST APIs to serve the data for web development. The results can be converted to other community standard dissemination formats, such as BioC (1) and brat standoff format[3].

B. Approaches to the Tool Integration Challenges

Dockerization: As text-mining tools may have different dependencies, we build a docker[4] image for each tool. This guarantees that the software is independent of the host machine and operating system, and can be run using a consistent command. Docker also creates an isolated environment for each running instance, allowing the tools to run in parallel without conflicting with each other.

Standardized JSON format: As different text-mining tools may use different output formats, we used one uniform JSON

[1] https://lucene.apache.org
[2] https://www.mongodb.com
[3] http://brat.nlplab.org/standoff.html
[4] https://www.docker.com

format for both input and output of the text-mining components. The basic schema is document-centric: each document contains a doc id field, text field, a list of properties, a hash-table of entities and a hash-table of relations. Each entity contains information such as entity type and offsets, while each relation contains relation type and its arguments.

Text alignment: As some text-mining tools may modify the original text during processing, we use Hirschberg's sequence alignment algorithm (2) to align the modified text and convert back to the original text. During the alignment process, only the character offset of an entity will be changed. After the alignment, the entity offsets in different text-mining results are based on the same text and ready to be merged.

III. Result

iTextMine currently consists of four in-house developed text-mining tools: (i) RLIMS-P (3) for mining protein phosphorylation (kinase-substrate-site), (ii) eFIP (4) for phosphorylation-dependent protein-protein interaction (PPI), (iii) miRTex (5) for miRNA-gene relation, and (iv) eGARD (6) for targeted therapy information from the scientific literature. For gene and other entity normalization, we incorporated results from PubTator (7).

For full-scale processing, we downloaded all Medline abstracts (June 2017) and ran the system pipeline to generate text-mining results. Table 1 summarizes the statistics of each tool—the number of positive abstracts, along with the counts of the specific relations types extracted. Overall, iTextMine identified 300,877 abstracts with at least one relation extracted by its underlying text mining tools.

TABLE I. iTextMine Medline abstract mining summary

Text Mining Tool	# Positive Abstracts		Relation types / counts
	Entities / Triggers	Entities + Relations	
RLIMS-P	289,258	264,163	phosphorylation (kinase-substrate-site): 454,389
eFIP	264,163	23,918	phosphorylation-dependent PPI: 38,814
miRTex	40,032	22,093	miRNA-target: 33,559 miRNA-gene regulation: 44,565 gene-miRNA regulation: 8,426
eGARD*	26,516	17,935	gene-disease-drug-response: 11,233

*The results are based on full-scale processing of 16.8 million abstracts for each tool, except eGARD which is still being processed with partial results only.

The website supports interactive query and display of text-mining results. User query to iTextMine will be sent as a query to PubMed to retrieve a list of PMIDs. Our system will then retrieve text-mining results from the database and generate a network using relations among entities. Redundant entities are merged if they are normalized to the same ID, or by the same text mention if not normalized.

We use the query "Triple negative breast cancer" as an example to demonstrate the integrated network with relations from RLIMS-P, eFIP, miRTex and eGARD (Fig 2). The interface also provides a document-centric view to display detailed text evidence (Fig 3).

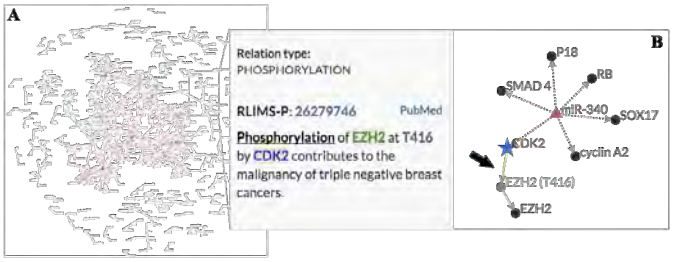


Fig. 2. Triple negative breast cancer network. A) The overall network contains 651 entities and 688 relations extracted by the four text-mining tools. B) Zoom in view. Different types of entity and relations are denoted by distinct node/edge colors and shapes. In this network, miR-340 (red triangle) regulates 5 genes (black circle) and a kinase (blue star). CDK2 phosphorylates EZH2 at Thr-416 and produces a proteoform (gray circle). By clicking the phosphorylation edge, the sentence describing the relation is shown.

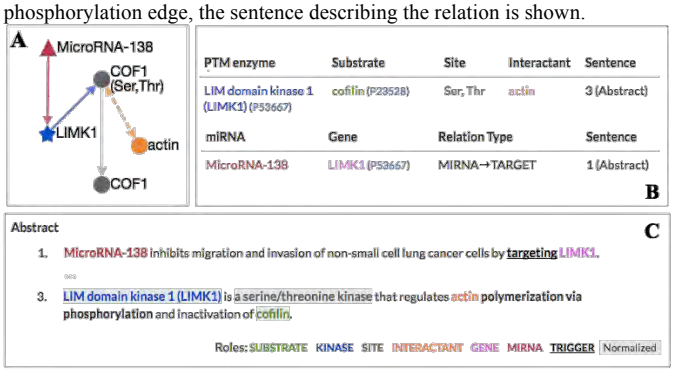


Fig. 3. Document-centric (PMID: 27665963) view. A) A network view summaries all entities and relations extracted from the abstract. RLIMS-P extracts a phosphorylation relation where kinase LIMK1 phosphorylates COF1; eFIP extracts PPI between the COF1 proteoform and actin; and miRTex extracts miRNA-gene regulation between MicroRNA-138 and LIMK1. B) Relation table lists relation arguments and attributes. C) Text evidence section displays the sentences with color-coded entities.

Acknowledgemet

This work was funded in part by grants from the National Institutes of Health (U01GM120953 and U01HG008390).

References

1. Comeau DC, Islamaj Dogan R, Ciccarese P, Cohen KB, Krallinger M, et al. (2013). BioC: a minimalist approach to interoperability for biomedical text processing. *Database* 2013, bat064.

2. Hirschberg DS. (1975). A linear space algorithm for computing maximal common subsequences. *Comm. ACM* 18, 341–343.

3. Torii M, Arighi CN, Li G, Wang Q, Wu CH, Vijay-Shanker K. (2015). RLIMS-P 2.0: A generalizable rule-based information extraction system for literature mining of protein phosphorylation information. *IEEE Transactions on Computational Biology and Bioinformatics* 12, 17–29.

4. Tudor CO, Ross KE, Li G, Vijay-Shanker K, Wu CH, Arighi CN. (2015). Construction of phosphorylation interaction networks by text mining of full-length articles using the eFIP system. *Database* 2015, bav020.

5. Li G, Ross KE, Arighi CN, Peng Y, Wu CH, Vijay-Shanker K. (2015). miRTex: A Text Mining System for miRNA-Gene Relation Extraction. *PLoS Computational Biology*, 11(9), e1004391.

6. Mahmood AS, Rao S, McGarvey PB, Wu CH, Madhavan S, Vijay-Shanker K. (2017) eGARD: Extracting associations between genomic anomalies and drug responses from text. *PLoS One* (in press)

7. Wei C-H, Kao H-Y, Lu Z. (2013). PubTator: a web-based text mining tool for assisting biocuration. *Nucleic Acids Research* 41, W518–W522.

Large-scale Automated Reading with Reach Discovers New Cancer Driving Mechanisms

Marco A. Valenzuela-Escárcega[1], Özgün Babur[2], Gus Hahn-Powell[3], Dane Bell[3], Thomas Hicks[1], Enrique Noriega-Atala[4], Xia Wang[5], Mihai Surdeanu[1], Emek Demir[2], Clayton T. Morrison[4]

[1]Dept. of Computer Science, University of Arizona, Tucson, USA
[2]School of Medicine, Oregon Health and Sciences University, Portland, USA
[3]Dept. of Linguistics, University of Arizona, Tucson, USA
[4]School of Information, University of Arizona, Tucson, USA
[5]Dept. of Molecular and Cellular Biology, University of Arizona, Tucson, USA

***Abstract*—PubMed, a repository and search engine for biomedical literature, now indexes more than 1 million articles each year. This exceeds the processing capacity of human domain experts, limiting our ability to truly understand many diseases. We present Reach, a system for automated, large-scale reading of biomedical papers that can extract mechanistic descriptions of biological processes with relatively high precision. We demonstrate that combining the extracted pathway fragments with existing biological data analysis algorithms that rely on curated models helps identify and explain a large number of previously unidentified mutually exclusive altered signaling pathways in seven different cancer types. This work shows that combining curated "big mechanisms" with extracted "big data" can lead to a causal, predictive understanding of cellular processes and unlock downstream applications.**

Keywords: machine reading, biological data analysis, hybrid human-machine models

In the period of 2004–2013, over 7.3 million journal articles were added to PubMed (1), and the rate is now over 1 million articles per year. Unfortunately, most of the mechanistic knowledge in the literature is not in a computable form and therefore remains hidden. Existing biocuration efforts are extremely valuable for solving this problem, but they are outpaced by the explosive growth of the literature. For example, we estimate that public pathway databases such as Pathway Commonscapture only 1–3% of the literature, and the gap widens every day.[1]

This gap severely limits the value of big data in biology. For example, some "driver" mutations in cancer exhibit a mutually exclusive pattern in a given cohort of patients. That is, the number of patients with both drivers will be smaller than what is expected by chance. This often happens because these alterations unlock the *same* cancer-driving pathways, and the positive selection of one diminishes substantially when the other is present. The Mutex algorithm (2) searches for groups of genes such that the alterations are mutually exclusive, and each gene in the group significantly contributes to this pattern. Pathway knowledge improves Mutex's accuracy by limiting the search space and reducing the loss of statistical power due to multiple hypothesis testing correction. It also provides mechanistic explanations of the observed correlations. Recall, however, can be low, due to the aforementioned database coverage issues. Researchers are thus faced with a choice between no-prior, high coverage methods without mechanistic explanations or low-coverage, prior-based methods that may overlook some key events.

We propose a natural language processing (NLP) approach that captures a system-scale, mechanistic understanding of cellular processes through automated, large-scale reading of scientific literature, and demonstrate that this approach leads to the discovery of novel biological hypotheses for multiple cancers. We call our approach Reach (REading and Assembling Contextual and Holistic mechanisms from text).

Reach is a hybrid statistical and rule-based approach, with its core consisting of compact grammars for the recognition of cellular processes. These grammars recog-

[1]Internal analysis of the Pathway Commons team.

[2]Inspired by NLP literature, we use "event" to indicate an interaction between multiple participants.

nize biological entities (e.g., genes, proteins, protein families, simple chemicals), events[2] that operate over these biochemical entities (e.g., biochemical reactions), and nested events that operate over other events (e.g., catalysis). These grammars were developed using the Odin information extraction framework (3–5). In all, we recognize 16 event types that follow the BioPAX representation (6), with a relatively small grammar of approximately 150 rules. This focus on grammar compactness is important for two reasons. First, it guarantees that the overall model is *interpretable*, i.e., it can be easily understood, modified, and extended by domain experts, i.e. biologists. And second, this compact grammar can be applied efficiently, permitting high-throughput processing.

The Reach architecture is implemented as a cascade of automata that recognize increasingly complex biomedical phenomena, as illustrated in Fig. 1.

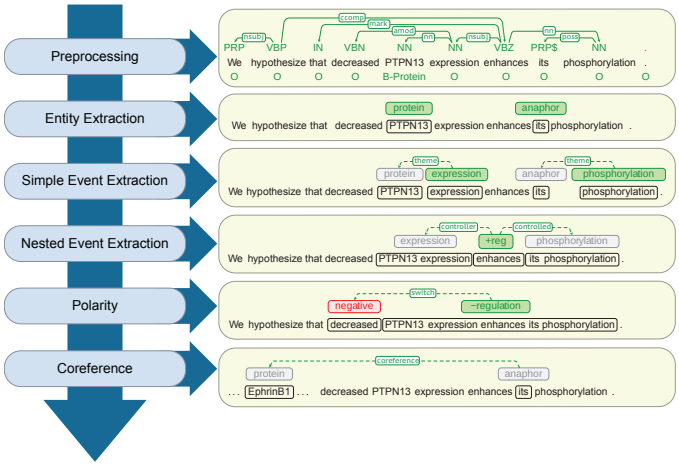


Fig. 1. A walkthrough example of the cascaded Reach architecture.

The Reach pipeline starts with preprocessing, splitting the text into sentences and further into words. Then, the text is annotated with parts of speech (e.g., `NN` indicates common noun), syntactic dependencies (e.g., `nsubj` captures the relation between a verb and its nominal subject), and named entity labels (e.g., `B-PROTEIN` indicates that the corresponding token is the beginning of a protein's name).

Next, Reach searches the preprocessed text. First, it extracts entity mentions (e.g., *PTPN13* as a `Protein` mention), including anaphoric mentions (e.g., *its*) to be resolved later. Second, Reach searches for events that operate directly on these entities, which we call simple events. These rules apply either over word (and part-of-speech) sequences or the dependency graph. For example, the *phosphorylation* event is extracted in this step. Third, Reach recognizes nested events, i.e., those with other events as arguments, such as the positive regulation in the example sentence. All three steps are implemented using an Odin grammar.

After the rule-based extraction, Reach applies further deterministic steps. It first detects event negation using the dependency graph, combining the negative *decreased* and positive *enhances* to produce a negative regulation. Then, it detects that the regulation is hypothesized using the dependency between *hypothesize* and the regulation. Finally, a deterministic coreference-resolution system (7) determines that *its* corefers with a previously mentioned entity *EphrinB1* (rather than *PTPN13*, for example).

In an independently administered evaluation[3], Reach was found to approach human precision at a throughput capable of reading the entire open source biomedical literature within days. Participating systems extracted mechanistic information from a thousand papers about the Ras signaling pathway over the course of a week. Three metrics were used to evaluate the participating systems: *throughput*, the estimated number of interactions produced per day; *generous precision*, the proportion of interactions that were considered useful by the expert panel; and *strict precision*, the proportion of interactions that were completely correct. Four consortia, each one potentially containing multiple teams, participated in the evaluation. Team 4 was a consortium between Reach and another group (4(B)). The results are summarized in Table I.

TABLE I. BIG MECHANISM EVALUATION

Team	Throughput	Strict Precision	Generous Precision
Team 1	110	54.69%	62.50%
Team 2	975	35.24%	42.07%
Team 3	242	56.76%	75.68%
Team 4	944	50.36%	66.42%
Reach	760	58.76%	74.23%
Team 4(B)	189	30.00%	47.50%

Reach-extracted pathway fragments improve the inference capacity of existing biological data analysis algorithms that already benefit from large curated models ("big mechanisms"). Specifically, we extended the Pathway Commons[4] human-curated pathways with fragments extracted by Reach from all papers in the Open Access subset of PubMed (1,046,662 papers as of June 2015) (Fig. 2).

[3]Conducted in the DARPA Big Mechanism program (`www.darpa.mil/program/big-mechanism`).
[4]`http://www.pathwaycommons.org/`

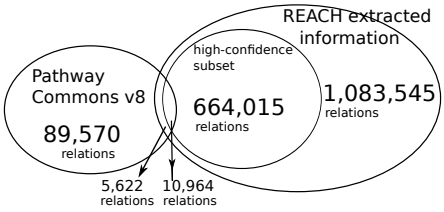


Fig. 2. The Reach output is about 12 times larger than the size of Pathway Commons. We conjecture that the small overlap is caused by the fact that the Reach interactions are extracted from open-access publications, whereas Pathway Commons pathways come mostly from other, paywalled publications. The high-confidence subset is of relations that were found in more than one paper.

Using this combined prior network we were able to identify previously unidentified, but highly statistically significant mutually exclusively altered signaling modules in TCGA cancer datasets using the Mutex algorithm described above. Fig. 3 shows Mutex groups for TCGA breast cancer, and Table II summarizes the findings for all enhanced cancer studies in TCGA. R represents the Mutex configuration using the combined Reach + Pathway Commons network, P the Mutex configuration using only the Pathway Commons network, and W the Mutex configuration uninformed by any supporting network.

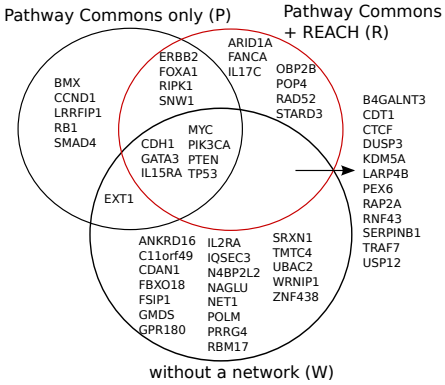


Fig. 3. Using Reach-extracted information allows Mutex to detect 7 new "driver" genes for breast cancer which are not detected using Pathway Commons only or without using any network. We observed similar results for multiple cancers in the TCGA dataset.

A manual evaluation of these modules by an external cancer researcher reveals that, despite the inherent noise in machine reading, 65% of the hypotheses proposed by Mutex+Reach are indeed correct according to the literature. Further, a simple redundancy filter that keeps Reach extractions only if they are seen at least twice in the literature increased this accuracy to 80%. This demonstrates that our approach systematically and incrementally increases coverage of prior, curated networks using NLP strategies, and, we believe, is valuable for molecular tumor boards and other cases where one needs to combine system-scale data with the knowledge in the literature.

TABLE II. MUTEX+REACH ANALYSIS OF TCGA

Cancer study	*R*	P	W	*R* − P − W	*R*W − P
BLCA	2	2	6	0	0
BRCA	30	17	40	7	12
CESC	5	6	7	0	0
DLBC	0	5	0	0	0
GBM	23	14	40	3	7
HNSC	26	23	25	3	2
KICH	0	0	6	0	0
LAML	2	2	2	0	0
LGG	26	12	51	0	14
LIHC	12	17	16	0	0
LUAD	14	16	11	1	0
OV	7	11	7	2	0
PAAD	22	7	17	10	5
SARC	15	22	25	0	0
THCA	9	11	12	0	0
UVM	2	3	34	0	0

REFERENCES

1. Vardakas, K.Z., Tsopanakis, G., Poulopoulou, A. and Falagas, M.E. (2015) An analysis of factors contributing to pubmed's growth. *Journal of Informetrics*, **9**(3), 592–617.

2. Babur, Ö., Gönen, M., Aksoy, B.A., Schultz, N., Ciriello, G., Sander, C. and Demir, E. (2015) Systematic identification of cancer driving signaling pathways based on mutual exclusivity of genomic alterations. *Genome biology*, **16**(1), 45.

3. Valenzuela-Escárcega, M.A., Hahn-Powell, G., Hicks, T. and Surdeanu, M. (2015) A domain-independent rule-based framework for event extraction. In *Proceedings of the 53rd Annual Meeting of the ACL: Software Demonstrations*. pp. 127–132.

4. Valenzuela-Escárcega, M.A., Hahn-Powell, G. and Surdeanu, M. (2015) Description of the Odin event extraction framework and rule language. *CoRR*, **abs/1509.07513**.

5. Valenzuela-Escárcega, M.A., Hahn-Powell, G. and Surdeanu, M. (2016) Odin's runes: A rule language for information extraction. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*. pp. 322–329.

6. Demir, E., Cary, M.P., Paley, S., Fukuda, K., Lemer, C., Vastrik, I., Wu, G., D'Eustachio, P., Schaefer, C., Luciano, J. et al. (2010) The BioPAX community standard for pathway data sharing. *Nature Biotechnology*, **28**(9), 935–942.

7. Bell, D., Hahn-Powell, G., Valenzuela-Escrcega, M.A. and Surdeanu, M. (2016) Sieve-based coreference resolution in the biomedical domain. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*. pp. 177–183.

POSTER SESSION

1. SPRENO: A BioC Module For Recognizing and Normalizing Species and Their Model Organisms (Onkar Singh and Hong-Jie Dai)
2. KinDER: A Biocuration Tool for Extracting Kinase Knowledge from Biomedical Literature (Adam Morrone and Daniel Dopp)
3. BELMiner – Information extraction system to extract BEL relationships (Ravikumar Komandur Elayavilli, Majid Rastegar-Mojarad and Hongfang Liu)
4. Generating Biological Expression Language Statements with Pipeline Approach and Different Parsers (Po-Ting Lai, Ming-Siang Huang, Wen-Lian Hsu, Richard Tzong-Han Tsai)
5. Automatic Extraction of BEL-Statements based on Neural Networks (Mehdi Ali, Sumit Madan, Asja Fischer, Henning Petzka and Juliane Fluck)
6. MayoNLP at the BioCreative VI PM Track: Entity-enhanced Hierarchical Attention Neural Networks for Mining Protein Interactions from Biomedical Text (Ravikumar Komandur Elayavilli, Majid Rastegar-Mojarad)
7. Mining protein interactions affected by mutations using a NLP based machine learning approach (Jinchan Qu and Albert Steppi)
8. Identifying Mutation-induced Protein-Protein Interactions in Scientific Literature (Chen-Kai Wang)
9. Extracting protein-chemical compound interactions from literature (Pei-Yau Lung, Tingting Zhao, Zhe He, Jinfeng Zhang)
10. SciLite - a platform to bridge scientific articles and biological data. (Aravind Venkatesan, Senay Kafkas, Julien Gobeill, Jee-Hyub Kim,
    Francesco Talo, Michele Ide-Smith, Patrick Ruch, Johanna McEntyre)
11. MecCog: A web-based framework for describing biological mechanism. (Kunal Kundu, Lipika R. Pal, Lindley Darden, John Moult)
12. Inter-annotator agreement and the upper bound on system performance in biomedical and general domain natural language processing. (Mayla Boguslav and Kevin Cohen)
13. A semi-supervised system for ontology enrichment. (Elizabeth T. Hobbs, Matthew Koert, Patrick K. O'Neill, Ivan Erill)
14. Biomedical Information Extraction at the SIB/BioMeXT group (Fabio Rinaldi)
15. Fast online entity recognition: OntoGene/BioMeXT in the BeCalm challenge (Lenz Furrer, Fabio Rinaldi)
16. Digital assisted curation to the rescue of traditional literature curation. (Fabio Rinaldi, Socorro Gama-Castro, Yalbi Itzel Balderas-Martínez, Oscar Lithgow, Hilda Solano, Mishael Sánchez-Pérez, Alejandra Lopez-Fuentes, Luis José Muñiz Rascado, Cecilia Ishida-Gutiérrez, Carlos-Francisco Méndez-Cruz, Alberto Santos-Zavaleta, Julio Collado-Vides)