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.