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.

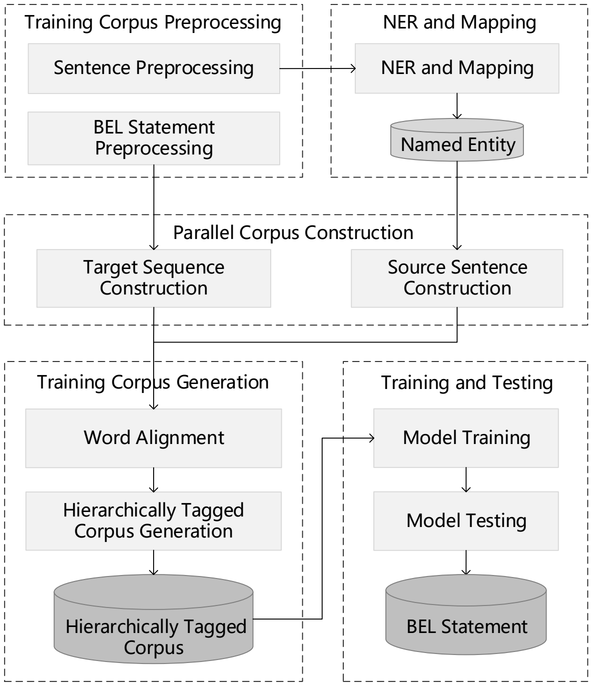
Training Corpus Preprocessing
Sentence Preprocessing
BEL Statement Preprocessing
NER and Mapping
NER and Mapping
Named Entity
Parallel Corpus Construction
Target Sequence Construction
Source Sentence Construction
Training Corpus Generation
Word Alignment
Hierarchically Tagged Corpus Generation
Hierarchically Tagged Corpus
Training and Testing
Model Training
Model Testing
BEL Statement

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]$_{\text{CHEM}}$ prevented [association of [GENE1]$_{\text{GENE}}$ and [GENE2]$_{\text{GENE}}$]$_{\text{complex}}$]$_{\text{decreases}}$”, 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.