Journal of Engineering Science and Technology Review 8 (4) (2015) 51-55

JOURNAL OF
Engineering Science and Technology Review

www.jestr.org

Research Article

Extracting Nested Biomedical Entity Relations by Tagging Dependency Chains

Xiaomei Wei [1,2], Yu Huang[2], Chen Lyu [1,3], and Donghong Ji [1,*]

[1] Computer School, Wuhan University, Wuhan 430072, China
[2] College of informatics, Huazhong Agriculture University, Wuhan 430070, China
[3] Singapore University of Technology and Design, 138682, Singapore

Received 24 June 2015; Accepted 14 October 2015

Abstract

Biomedical event extraction is an important research topic in the field of biomedical text mining. However, much research work is required before event extraction systems become applicable. Thus, we proposed a novel and efficient approach for extracting nested biomedical events. First, using dependency parsing, we extracted the target sequences that contained biomedical entity (trigger/argument) chains. Second, the Condition Random Fields (CRFs) model was used to tag the entity chains which represented the nested argument-trigger edges. Thirdly, the post-processing step was used to output the events. This method is a new attempt to treat the biomedical event extraction as a sequence tagging problem. The experiment results showed that we got the performance of 47.3 in F-score which is promising when compared with the joint ML-based system in BioNLP-ST2013. Furthermore, we estimated the results of the trigger detection, which outperformed the state-of–the-art systems on the same corpus. Therefore, our work is a positive contribution to the biomedical text mining community.

Keywords: Joint; Event extraction; Entity chain; Dependency; Tag

1. Introduction

Biomedical event extraction has become an important research topic in the field of biomedical natural language processing in recent years [1]. Biomedical events describe the fine-grained relations among biomedical entities. The biomedical literature contains substantial information regarding relations among biomedical entities, and these relations must be extracted to construct a knowledge database for researchers. This effort led to the BioNLP GE shared task (BioNLP-ST, hereafter) series [2-4], which aims to extract nested bio-molecular events from biomedical text. BioNLP-ST addressed nine types of biomedical molecular events related to protein biology. These events can be grouped into three categories: Simple, Binding, and Regulation. Simple events (Gene_expression, Transcription, Protein_catabolism, Phosphorylation, Localization) take one protein argument. Binding events (Binding) have one or more protein arguments. Regulation events (Positive_regulation, Negative_regulation and Regulation) have one obligatory Theme and one optional Cause argument. Each argument of Regulation events could be either a protein or another event. A Regulation event is considered nested if it has another event as its argument. A sample of an event annotation of a sentence (Sen.1) from training corpus is illustrated in Fig. 1.

Sen.1: BMP-6 did not induce significant changes in the protein expression of Id2 and Id3.

In this sentence, the trigger words are presented in bold font, whereas the protein arguments are expressed in underline font. In the definition of BioNLP09-ST [2], both triggers and arguments are called entities. In the upper textbox of the figure, proteins “BMP-6”, “Id2”, and “Id3” are labeled as T73, T74, and T75, respectively. In the lower textbox, T50 and T51 are two labels of triggers, and E27 and E28 are two events.

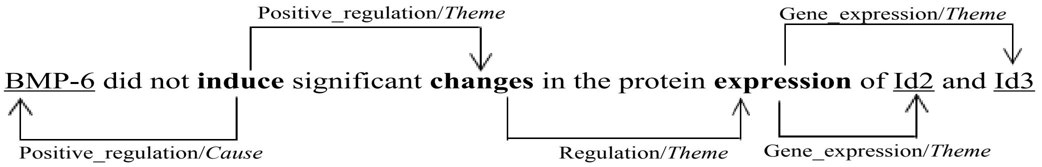
Biomedical event extraction is a complex task that requires study before being applied. The complexity of event extraction rests on two aspects. First, the sentences in the biomedical literature are typically very complex. Second, many biomedical events are nested and are thus different from the event definition in the common field, such as the ACE2005 [5] event task. As shown in Fig.1, event E79 contains the trigger word T169 and the protein argument T74. Meanwhile, event E79 is the argument of another event E76. Therefore, event E76 is a nested event while it is the argument of event E75. When multiple nested layers exist, extracting events becomes more difficult because errors in the lower layers could lead to errors in the upper layers.

2. Related works

To date, researchers have proposed many experimental methods to extract biomedical event based on

[*] E-mail address: may@mail.hzau.edu.cn

ISSN: 1791-2377


Positive_regulation/*Theme*
Gene_expression/*Theme*
BMP-6 did not **induce** significant **changes** in the protein **expression** of Id2 and Id3
Positive_regulation/*Cause*
Regulation/*Theme*
Gene_expression/*Theme*

```
T73     Protein 4993 4998      BMP-6      T167   Positive_regulation 5007 5013    induced
T74     Protein 5063 5066      Id2        T168   Regulation 5026 5033            changes
T75     Protein 5071 5074      Id3        T169   Gene_expression 5049 5059  expression
                                          E75    Positive_regulation:T167 Theme:E76 Cause:T73
                                          E76    Regulation:T168 Theme:E79
                                          E77    Positive_regulation:T167 Theme:E78 Cause:T73
                                          E78    Regulation:T168 Theme:E80
                                          E79    Gene_expression:T169 Theme:T74
                                          E80    Gene_expression:T169 Theme:T75
```

Fig. 1. An example sentence with gold event annotations

the datasets of BioNLP-ST. These approaches can be divided into two main groups: pipeline-based and joint. The first group [6–9] generally treats the event extraction procedure as a pipeline of multiple steps, i.e., identifying event triggers and then assigning their arguments. For this group, the errors from previous steps propagate into subsequent steps. To overcome this issue, the second group adopted joint learning models [10-13]. Using the joint model, Riedel S. et al. achieved the best F-score on BioNLP-ST2011 [11]. The latest event extraction systems have likewise adopted the joint model [12] and have demonstrated good performance. However, joint models suffer from the complexity of inference [13]. To remain tractable, joint models usually need to represent the task in a simplified way, which makes certain features harder to capture.

In this paper, we introduced a novel ML-based method based on the Condition Random fields (CRFs) model to extract biomedical events jointly. This method does not need complex inference and task decomposition, and it includes only three steps. First, basing on syntactic dependencies, we extracted the target sequences that contained entity chains. In the chain, the current entity acts as a trigger of the previous one and as argument of the next. Therefore, the entity chain was composed of nested argument–trigger pairs, as shown in Fig. 1. Event E79 contains the trigger word T169 and the protein argument T74, and the role of the argument is Theme. Meanwhile, event E79 is the argument of another event E76, and its role is Theme, and so on. Thus, the triggers in these nested events form a chain: “expression” << “changes” << “induce” (corresponding label sequence: T169 << T168 << T167). Second, the CRFs model was used to tag entity chains. Third, on the basis of the tagged entities chains, a pro-processing step was implemented to construct the nested events. When evaluated on the BioNLP-ST2013 dataset, our system performance achieved an F-score of 47.3, outperforming the baseline system that also used joint ML-based approach on BioNLP-ST2013. In addition, we evaluated the results of the trigger detection subtask on the same corpus and obtained a promising improvement when compared with similar systems.

3. Method

The workflow of the biomedical event extraction system is shown in Fig. 2.

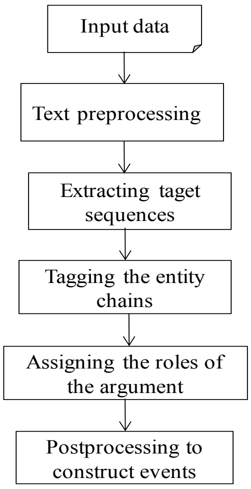

Input data
Text preprocessing
Extracting taget sequences
Tagging the entity chains
Assigning the roles of the argument
Postprocessing to construct events

Fig.2. Overview of the extracting system

3.1 Pre-processing

3.1.1 Constructing the trigger dictionary

In our method, all of the words that have acted as triggers were called candidate triggers. For each trigger, we counted nine frequencies corresponding to the nine types. We also counted the frequency in which each candidate trigger acted as common words (not a trigger in the sentence) in the training data. On the basis of the candidate trigger list and their frequencies, the trigger dictionary was built. In the trigger dictionary, each item contains the candidate trigger, its lemma, the counted frequencies that the word appeared as event triggers, and the frequency that it appeared as common words.

3.1.2 Replacing the protein names

We replaced all the names of annotated protein entities with label “PROi”, where “i” represents the order that the annotated protein appeared in the text. If two labels were connected with each other by a comma or a conjunction (such as “PROi_ and PROj_”), we combined them and replaced the combination with one label. For example, if “PRO1, PRO2, and PRO3” appears in a sentence, it will be replaced with “PRO1_PRO2_PRO3_”. After this pre-processing, these proteins will be treated as one word and obtain the same parsing result. This treatment plays positive contribution to the extraction because it combines the tokens that have the same semantics while it doesn’t change the syntactic structure of the sentence. The replacement for the example passage is shown in Fig.3(a).

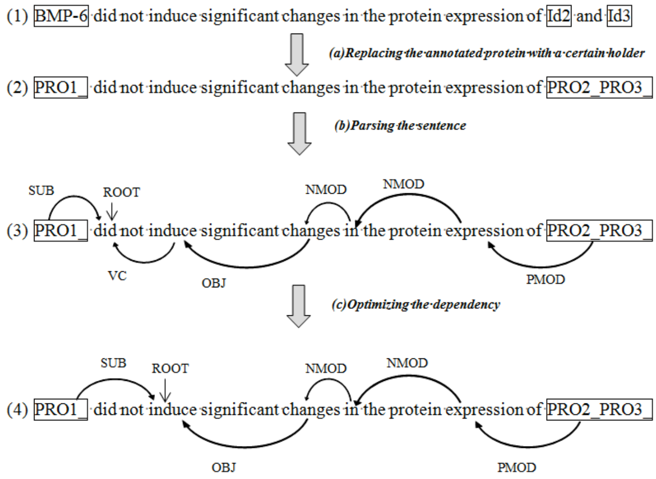

(1) BMP-6 did not induce significant changes in the protein expression of Id2 and Id3
(a)Replacing the annotated protein with a certain holder
(2) PRO1_ did not induce significant changes in the protein expression of PRO2_PRO3_
(b)Parsing the sentence
SUB
ROOT
NMOD
NMOD
(3) PRO1_ did not induce significant changes in the protein expression of PRO2_PRO3_
VC
OBJ
PMOD
(c)Optimizing the dependency
SUB
ROOT
NMOD
NMOD
(4) PRO1_ did not induce significant changes in the protein expression of PRO2_PRO3_
OBJ
PMOD

Fig.3. Preprocessing of Sen.1 (a) Replacing the proteins with certain holders; (b) Parsing Sen.1; (c) Optimizing the parse output of the verb phrase.

3.1.3 Parsing sentences and trimming the parse output

Similar with other text mining approaches employing dependency parsing [14-16], the dependency graph or syntactic parse tree of a candidate sentence is known to carry valuable information for event extraction tasks. Our method also depends closely on the dependency parsing. Before the text was sent to the parser, we removed the sentences without annotated proteins and segmented the rest. The sentences containing protein entities were incorporated to Gdep, which is the parser for biomedical text. The parse output is shown in Fig.3(3). Although the Gdep parser is specific for biomedical text, some outputs are still inefficient in extracting the edge of trigger-argument. Thus, we trimmed the output of the parser for the extraction of the semantic dependency sequences. The optimization strategies include the following two steps:

1) Re-assigning the head word of the verb phrase

In the dependency structure shown in Fig.3(3), the root node (also head word of the verb phrase) is the word “did” and the dependency path from protein entity “PRO1_” to candidate trigger “induce” is “PRO1_>> did << induce”, where both “PRO1_” and “induce” depend on the word “did”. Here “A>>B” represents that “A” depends on “B” and “A<<B” is in reverse. Using our optimization strategies the head word of the verb phrase is “induce” and the dependency relation of the trigger-argument is “PRO1_ >> induce”, in which the two entities connect directly, as shown in Fig.3(4).

2) Combining the predicative structure

Most of the triggers are nouns, verbs, and adjectives, except for a few other parts of speech. In predicative structure, usually it is the adjective that acts as the trigger. However, in the dependency relation of the predicative structure, the adjective depends on the auxiliary verb, which is not intuitive to extract the entity chains. Therefore, we combined the adjective phrase into the verb phrase and assigned the adjective as headword of the new verb phrase.

3.2 Extracting the target sequences

In a nested biomedical event, the entities from the bottom (protein entities) to the top argument trigger form a chain. The entity in the middle of the chain is the candidate trigger of the previous entity, and it is the argument of the entity behind it. In the sentence dependency graph, most of the chains are contained on the dependency path from protein node to root. We extracted the target sequences based on the dependency parsing.

For sen.1 in Fig.2(4), we extracted two sequences from the proteins to the root node as follows:

Seq1: *PRO1_>>**induce***
Seq2:*PRO2_PRO3_>>of>>**expression**>>in>>**changes**>>**induce***

In the target sequences Seq1 and Seq2 above, the words in bold are the trigger entities that should be extracted. The nested entity pairs in the sequences form the entity chains. However, although the majority of nested argument-trigger pairs are contained in the target sequences, some pairs contained in special syntactic structure, such as compound words and coordinate structures, will be omitted.

For example as following passage:

Sen.2*: ...to **detect** PRO1_-**induced changes** in the **phosphorylation** of PRO2_.....*

In the compound word “PRO1_-induced”, there is a trigger-argument pair (induced, PRO1_). It is necessary to decompose the compound word and extract the dependency sequences as follows:

Seq3:*PRO2_>>of>>**phosphorlation**>>in>>**changes**>> **induced**>>**detect***
Seq4*: PRO1_>>**induced***

Coordinate structure makes it difficult to extract the target sequences directly. In our experiment, it is necessary to split the coordinative constituent to extract the target sequences. Consider the following example:

Sen.3*: ...PRO1_ **induces** proteasomal **degradation** of PRO2_ and **inhibits** PRO3_ activity...*

In this passage, the coordinated words “induces” and “inhibits” have the common subject which is the protein entity “PRO1_”. We split the coordinated structure and extracted the sequences as follows:

Seq5: *PRO1_>>**induces***
Seq6: *PRO1_>>**inhibits***
Seq7: *PRO2_>>of>>**degradation**>>**induces***
Seq8: *PRO3_>>**inhibits***

3.3 Tagging the entity chains in the target sequences

We used the conditional random fields (CRFs) model [17] to tag the trigger entity chains in the target sequences with the pre-defined labels. The CRFs were first introduced by Lafferty et al., which represented a form of undirected graphical model. CRFs provide a discriminative framework for building structured models to segment and label sequence data. Recently, there has been increasing interest in CRFs, with successful applications including text processing [18, 19] and bioinformatics [20]. Specifically, we use the CRF++ toolkit, which has been shown to be highly successful over various chunking tasks. The algorithm is described in the following paragraphs.

Table 1. P-thresholds for each type of triggers

Trigger types	Gen	Tra	Pro	Pho	Loc	Bin	Reg	Pos	Neg
P-thresholds	0.25	0.25	0.25	0.25	0.25	0.08	0.3	0.25	0.16

Assume that we have an input sequence of observations x = (x_1, x_2 …, x_n), and a state variable y = (y_1, y_2, … , y_n) that needs to be inferred from the given observations. Given x, the probability of a label sequence y determined by CRF model is calculated as follows

$$p(y \mid x, \lambda) = \frac{1}{Z(x)} \exp(\sum_j \lambda_j F_j(y, x)) \qquad (1)$$

where λ_j is the parameter to be estimated from training data and indicates the informativeness of the respective feature Z(x) is a normalization factor. $F_j(y, x)$ in formula (1) is calculated as

$$F_j(y, x) = \sum_{i=1}^{n} f_j(y_{i-1}, y_i, x, i) \qquad (2)$$

where each $f_j(y_{i-1}, y_i, x, i)$ is a state function.

In order to tag the entity chains, we defined label sets = {Gene_expression, Transcription, Protein_catabolism, Phosphorylation, Localization, Binding, Regulation, Positive_regulation, Negative_regulation, NONE}, which were composed of nine types of labels of the trigger/event plus the label "NONE". The type labels were used to tag the triggers, and the label "NONE" was used to tag the common words in the dependency sequences. The features used in the CRFs model template were Lemma, POS (part of speech), dependency labels, and the relative position of a word in a sentence (left or right).

For example, using our CRFs model, the dependency sequences Seq1 and Seq2 in section 3.2 were tagged with the sequences Tag1 and Tag2, as follows:

Seq1: *PRO1_>>**induce***
Tag1: NONE>>**Positive_regulation**
Seq2:*PRO2_PRO3_>>of>>**expression**>>in>>**changes** >>**induce***
Tag2:NONE>>NONE>>**Gene_expression**>>NONE>> **Regulation**>> **Positive_regulation**

3.4 Revising the output of the CRFs

After tagging, two steps were implemented to improve the output of the CRFs. In sequences extraction, some trigger entities can be contained by different dependency sequences. For example, both Seq5 and Seq7 contained the entity "induces", which can be tagged as different labels. When CRFs tagged the same entity as different labels, we compared their probability values from CRFs and chose the label with higher marginal probability as the tagging result. In addition, ML-based approaches tend to classify all the data into the majority class [21, 22]. In biomedical texts, a candidate trigger often appears as a common word, so the entities tend to be tagged as "NONE" by the discriminative probabilistic CRFs model. Accordingly, we revised the output of CRFs by setting the marginal probability thresholds. The probability threshold P was learned from the developed data and obtained nine values related to nine types of triggers, as shown in Table 1. According the rule of revision, if the word is tagged as "NONE" and the second-highest probability is larger than P, the tagger of this word should be corrected with the label corresponding to the second-highest probability. If the word is not in the dictionary, it should be ignored.

3. 5 Post-processing to output the events

Based on the dependency sequences and their label sequences above, we have extracted the entity pairs, their labels, and the role labels to form the event edge tetrads, shown as follows:

Positive_regulation(PRO1_, induce, Cause)
Gene_expression(PRO2_PRO3_, expression, Theme)
Regulation(expression, changes, Theme)
Positive_regulation(changes, induce, Theme)

Based on the tetrads above, it is easy to construct the event output according to the format specification of BioNLP2013-ST. Simple events have only one Theme argument represented by proteins. The Binding events have one or more Theme arguments which are all proteins. The regulation events are the most complex because they have one obligatory Theme and one optional Cause argument. In addition, the argument of the regulation events could have the other events as their arguments. In a tetrad, if the first entity is a trigger, it must be a Regulation event or it would be removed. Based on the tetrads above, we obtained the formal event representation as follows:

*E1 Gene_expression:expression Theme:PRO2_*
*E2 Gene_expression:expression Theme:PRO3_*
E3 Regulation:changes Theme:E1
E4 Regulation:changes Theme:E1
*E5 Positive_regulation:induce Theme:E4 Cause:PRO1_*
*E6 Positive_regulation:induce Theme:E5 Cause:PRO1_*

4 Experiments

4.1 Data sets and evaluation criteria

The BioNLP2013-ST dataset include training, development, and test data. The training set and the development set are prepared with the gold protein annotation and the gold event annotation given. The testing set is provided without the gold event annotation. Table 2 shows the statistics of training, development, and test datasets [4].

Table 2. Statistics of training, development, and test datasets

Items	Training	Develop	Test
Articles	10	10	14
Words	54938	57907	75144
Proteins	3571	4138	4359
Entities	121	314	327
Events	2817	3199	3348

We adopted the Precision-Recall metric to evaluate the performance. Precision-Recall metric is defined as the following:

$$P = \frac{TP}{TP + FP}, \qquad R = \frac{TP}{TP + FN}, \qquad F = \frac{2P \times R}{P + R}$$

Here P (Precision) is the ability of a system to present only relevant items; R (Recall) is the ability of a system to present all relevant items; and F (F-measure) is the harmonic mean of precision and recall.

4.2 Results

Table 3 shows the results of our extraction method evaluated on the test dataset of BioNLP-ST2013. The performance of the method on test dataset was 47.3 F-score, which a 2.8 point lower that of the development dataset. It achieved the best results on simple events (SIMPLE ALL), with the F-score 73.47, followed by protein modification events (PROT-MOD ALL) with the F-score 67.04, and regulatory events (REGULATION-ALL) with the F-score 34.33. However, F-score of the binding events reached 42%, which is better than that of the development dataset. The difference may be attributed to the text materials chosen for the corpus.

Table 3. Results on test dataset of BioNLP-ST2013

Event Type	recall	prec.	fscore
Gene_expression	70.44	88.87	78.59
Protein_catabolism	53.47	75.00	62.43
Transcription	57.14	57.14	57.14
Localization	35.34	85.00	49.94
[SIMPLE ALL]	63.99	86.25	73.47
Binding	37.54	47.46	42.00
Phosphorylation	73.12	74.03	73.57
[PROT-MOD ALL]	61.26	74.03	67.04
Regulation	17.36	23.67	20.03
Positive_regulation	33.01	50.96	40.07
Negative_regulation	24.52	38.94	30.1
[REGULATION ALL]	28.4	43.39	34.33
[EVENT TOTAL]	40.2	57.44	47.3

Table 4 presents a comparison between the results of our method and the similar system. In BioNLP2013-ST, the participators adopted various methods to implement the task, including ML-based pipeline, Rule-based pipeline, Mixed-pipeline, Joint pattern, and Joint ML. Among these systems, HDS4NLP[4] was the best system that adopted the joint ML-based method to extract the events. Similar to this system, our method adopted the ML-based method CRFs to tag the triggers and argument-trigger edges jointly.

Table 4. Performance comparison with other systems in BioNLP-ST2013

Event types	Rec.	Prec.	F-score	Rec.	Prec.	F-score
SIMPLE ALL	75.27	83.27	79.07	63.99	86.25	73.47
Binding	41.74	33.74	37.32	37.54	47.46	**42.00**
PROT-MOD ALL	70.68	75.84	73.17	61.26	74.03	67.04
REGULATION ALL	16.67	30.86	21.64	28.4	43.39	**34.33**
EVENT ALL	37.11	51.19	**43.03**	40.2	57.44	**47.3**
Systems	HDS4NLP			Ours		

The comparison results in Table 4 showed that our method obtained >4% improvement in overall F-score. Our method also showed significant advantages on the Binding and Regulation events. However, in the simple (SIMPLE ALL) and protein modification (PROT-MOD ALL) events, the F-scores were lower than HDS4NLP system. This result conformed to the method we adopted. According to the biomedical event definition, only Regulation events could be nested. Our approach is based on sequence tagging model, and is appropriate for tagging chains containing multiple entities. Accordingly, our method showed good performance on regulation types.

In event extraction systems, the results of trigger detection could show the performance of event extraction in some extent. Recently, several approaches were proposed to focus on trigger detection subtask [23-25], which conducted experiments on BioNLP2009-ST corpus. Similar with these systems, we chose the BioNLP2009 training datasets as training sets, and the BioNLP2009-ST development datasets as test sets to implement the experiment. Also, we estimated the result of trigger detection on the BioNLP2013-ST. The experiment results and comparison with the state-of-the-art trigger detection systems are shown in Table 5.

Table 5. Performance comparison of Trigger detection on BioNLP2009-ST datasets

	Precision	Recall	F-score
Zhang[23]	79.83	56.02	65.84
Campos[24]	69.3	57.3	62.7
Ours	76.56	61.22	68.03

To test the robustness and reproducibility of this method, we also estimated the results on the dataset of BioNLP2013-ST using the same model and parameters mentioned above. Different from the corpus of BioNLP2009-ST which were retrieved from PubMed abstracts, the data set of BioNLP2013-ST were composed of PubMed full papers. Table 6 shows the results of the experiment on BioNLP2013-ST corpora. We obtained the F-score of 71.33, which is higher than that of BioNLP-ST2009. This may be attributed to the text material of the corpus. The result shows that our method also performs better on full paper text materials.

Table 6. Result estimation on the datasets of BioNLP2013-ST

	Precision	Recall	F-score
Trigger detection	80.54	64	71.33

4.3 Discussion

In our event extraction system, we used the sequence tagging method based on CRFs model to tag the relations of the trigger and argument, which is a new research attempt in this field. The results showed that our method achieved overall improvement compared with the baseline system. Meanwhile, the performance of the complex events (Binding and Regulation) showed significant advantage. This improvement was benefited from the dependency sequences containing the entity chains. The entity chains were formed by nested argument-trigger pairs. Therefore, our method improved the results of the nested Regulation events distinctly. In these chains, the entities are the context for each other, thus, they could identify each other instead of transmitting errors. However, while this strategy of extracting the dependency sequences was helpful to enhance the precision, it also reduced the recall.

In addition, there are several factors that affect the performance of the system. First, the performance of the parser is important to our method. Because the key step in our method is to extract the target dependency sequences, the output of the parser is crucial. However, the accuracy of most dependency parser is below 90%, which affects the accuracy of the extraction of target sequences. If the output of the parser is incorrect, the target sequences are unreliable and we cannot grasp the correct entity chains. Second, the coverage of the target sequences is important for this method. However, the biomedical text often contains long and complex sentence structures which make the target sequences difficult to extract. Therefore, some omitted sequences containing entity chains affected the recall of the overall system.

5. Conclusion

We presented a joint inference approach to extract the biomedical event of BioNLP-ST13. This approach simplified the task of extracting complex structure with regard to sequence tagging. This treatment avoided the complex inference but kept the structure feature of the entities. Both dependency features and context features were contained in the target sequences, which made it possible for a common machine learning model to extract the complex semantic relations. The evaluation on BioNLP-ST2013 test corpora has shown that our system achieved good results comparable with the similar system of BioNLP-ST2013. For further studies, we plan to enhance the recall of the event extraction and resolve the anaphora in the text. Moreover, combining the outputs of multiple parsers may further boost performance.

Acknowledgements

This work was supported by National Natural Science Foundation of China ((No. 61202304, 61133012), Major bidding project of the national social science research plans (11&ZD189), and Provincial Natural Science Foundation (NO. 012FFA088).

References

1. Miwa, M., Sætre, R., Kim, J. D., & Tsujii, J. I. "Event extraction with complex event classification using rich features." *Journal of bioinformatics and computational biology*, 8(01), 2010, pp.131-146.
2. Kim, J.D., Ohta, T., Pyysalo S., Kano, Y., Tsujii J., "Overview of BioNLP'09 shared task on event extraction," *Proceedings of BioNLP'09 Shared Task Workshop,* ACL, Morristown, NJ, USA, 2009, pp.1-9.
3. Kim, J.D., Wang. Y., Takagi, T., Yonezawa A., "Overview of Genia Event Task in BioNLP'09 shared task2011", *Proceedings of BioNLP Shared Task2011 Workshop, ACL*, Portland, Oregon, USA, June 24, 2011, pp. 7-15.
4. Kim, J.D., Wang, Y., Yasunori, Y., "The Genia Event Extraction Shared Task, 2013 Edition-Overview," *Proceedings of the BioNLP Shared Task 2013 Workshop*, Sofia, Bulgaria, August 9, 2013, pp. 8-15.
5. ACE2005. *Automatic Content Extraction 2005 Evaluation(ACE2005)*, http://itl.nist.gov/iad/mig/tests/ace/2005/
6. McClosky, D., Surdeanu, M., Manning, C. D, "Event extraction as dependency parsing", *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Portland, Oregon, June 19-24, 2011, pp.1626–1635.
7. Björne, J., & Salakoski, T, "TEES 2.1: Automated Annotation Scheme Learning in the BioNLP 2013 Shared Task," *Proceedings of the BioNLP Shared Task 2013 Workshop, ACL,* Sofia, Bulgaria, August 9, 2013, pp.16-25,
8. Hakala, K., Landeghem, S.V., Salakoski, T., Peer, Y.V., Ginter, F., "EVEX in ST'13: Application of a large-scale text mining resource to event extraction and network construction," *Proceedings of the BioNLP Shared Task 2013 Workshop, ACL*, Sofia, Bulgaria, 2013, pp.26-34.
9. Bui, Q.C., Sloot, P.M.A, "A robust approach to extract biomedical events from literature", *Bioinformatics*, 28(20), 2012, pp.2654-2661.
10. Lu, W., Roth, D., "Automatic Event Extraction with Structured Preference Modeling," *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*. Stroudsburg, PA, USA, 2012, pp.835-844.
11. Riedel, S., McClosky, D., Surdeanu, M., McCallum, A., Manning, C.D., "Model Combination for Event Extraction in BioNLP 2011," ACL, Stroudsburg, PA, USA, 2011, pp.51-55.
12. Li, Q., Ji, H., Huang, L., "Joint Event Extraction via Structured Prediction with Global Features," *Proceedings of the 52th Annual Meeting of the Association for Computational Linguistics*, Sofia, Bulgaria, 2014, pp.73-82.
13. Poon, H., Vanderwende, L., "Joint Inference for Knowledge Extraction from Biomedical Literature." *In Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Los Angeles, California, 2010, pp.813-821.
14. Björne, J., Ginter, F., Pyysalo, S., Tsujii, J., Salakoski, T., "Complex event extraction at PubMed scale." *Bioinformatics*, 26(12), 2010, pp. 382-390.
15. Giles, C. B., & Wren, J. D., "Large-scale directional relationship extraction and resolution." *BMC Bioinformatics,* 9(Suppl 9), 2008, S11, pp.1-13.
16. M. Miwa, P. Thompson, S. Ananiadou, "Boosting automatic event extraction from the literature using domain adaptation and coreference resolution," *Bioinformatics*, 28(13), 2012, pp.1759-1765.
17. Lafferty, J.D., McCallum, A., Pereira, F.C.N., "Conditional random fields: probabilistic models for segmenting and labeling sequence data." *In International Conference on Machine Learning*, Morgan Kaufmann, 2001, pp. 282-289.
18. Nakagawa, T., Inui, K., Kurohashi, S., "Dependency Tree-based Sentiment Classification using CRFs with Hidden variables," *Human Language Technologies:The 2010 Annual Conference of the North American Chapter of the ACL*, 2010, pp. 786-794.
19. Settles, B., "Abner: an open source tool for automatically tagging genes, proteins, and other entity names in text," *Bioinformatics*, 21(14), 2005, pp. 3191-3192.
20. Sato, K., & Sakakibara, Y. "RNA secondary structural alignment with conditional random fields." *Bioinformatics*, 21(suppl 2), 2005, pp.ii237-242.
21. M. A. Maloof. "Learning When Data Sets are Imbalanced and When Costs are Unequal and Unknown," *In ICML-2003 workshop on learning from imbalanced data sets II*, Washington DC, 2003, pp. 1-8.
22. He, H., Garcia, E. A., "Learning from Imbalanced Data." *IEEE Transactions on Knowledge and Data Engineering,* 21(9), 2009, pp. 1263-1284.
23. Zhang, Y., Lin, H., Yang, Z., Wang, J., & Li, Y, "Biomolecular event trigger detection using neighborhood hash features," *Journal of theoretical biology*, 318, 2013, pp. 22-28.
24. Campos D., Bui Q. C., Matos S., et al. "TrigNER: automatically optimized biomedical event trigger recognition on scientific documents," *Source code for biology and medicine*, 2014, 9(1), pp.1-13.
25. Martinez D., Baldwin T., "Word sense disambiguation for event trigger word detection in biomedicine," *BMC bioinformatics*, 2011, 12(Suppl 2), S4, pp.1-8.