A Multi-Phase Approach to Biomedical Event Extraction

Hyoung-Gyu Lee, Han-Cheol Cho, Min-Jeong Kim
Joo-Young Lee, Gumwon Hong, Hae-Chang Rim
Department of Computer and Radio Communications Engineering
Korea University
Seoul, South Korea
{hglee,hccho,mjkim,jylee,gwhong,rim}@nlp.korea.ac.kr

Abstract

In this paper, we propose a system for biomedical event extraction using multi-phase approach. It consists of event trigger detector, event type classifier, and relation recognizer and event compositor. The system firstly identifies triggers in a given sentence. Then, it classifies the triggers into one of nine pre-defined classes. Lastly, the system examines each trigger whether it has a relation with participant candidates, and composites events with the extracted relations. The official score of the proposed system recorded 61.65 precision, 9.40 recall and 16.31 f-score in approximate span matching. However, we found that the threshold tuning for the third phase had negative effect. Without the threshold tuning, the system showed 55.32 precision, 16.18 recall and 25.04 f-score.

1 Introduction

As the volume of biomedical literature grows exponentially, new biomedical terms and their relations are also generated. However, it is still not easy for researchers to access necessary information quickly since it is lost within large volumes of text. This is the reason that the study of information extraction is receiving the attention of biomedical and natural language processing (NLP) researchers today.

In the shared task, the organizers provide participants with raw biomedical text, tagged biomedical terms (proteins), and the analyzed data with various NLP techniques such as tokenization, POS-tagging, phrase structure and dependency parsing and so on. The expected results are the events, which exist in the given text, consisting of a trigger and its participant(s) (Kim et al., 2009).

The proposed system consists of three phases; event trigger detection phase(TD phase), event type classification phase(TC phase), relation recognition and event composition phase(RE phase). It works in the following manner. Firstly, it identifies triggers of a given biomedical sentence. Then, it classifies triggers into nine pre-defined classes. Lastly, the system finds the relations between triggers and participant candidates by examining each trigger whether it has relations with participant candidates, and composites events with the extracted relations. In the last phase, multiple relations of the same trigger can be combined into an event for *Binding* event type. In addition, multiple relations can be combined and their participant types can be classified into not only *theme* but also *cause* for three *Regulation* event types.

In this paper, we mainly use dependency parsing information of the analyzed data because several previous studies for SRL have improved their performance by using features extracted from this information (Hacioglu, 2004; Tsai et al., 2006).

In the experimental results, the proposed system showed 68.46 f-score in TD phase, 85.20 accuracy in TC phase, 89.91 f-score in the initial step of RE phase and 81.24 f-score in the iterative step of RE phase, but officially achieved 61.65 precision, 9.40 recall and 16.31 f-score in approximate span matching. These figures were the lowest among twenty-four shared-task participants. However, we found that the threshold tuning for RE phase had caused a negative effect. It deteriorates the f-score of the

Proceedings of the Workshop on BioNLP: Shared Task, pages 107–110,
Boulder, Colorado, June 2009. ©2009 Association for Computational Linguistics

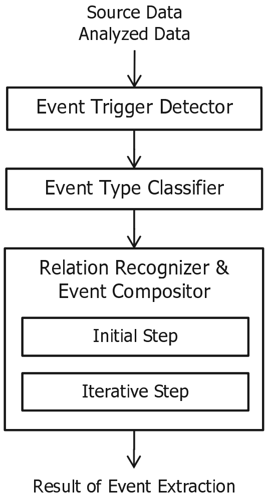

Figure 1: System Architecture

proposed system by enlarging the gap between precision and recall. With the default threshold, the system showed better result in the final test data, 55.32 precision, 16.18 recall and 25.04 f-score with the rank 17th among 24 teams.

2 System Description

Figure 1 shows our bio-event extraction system which consists of Event Trigger Detector, Event Type Classifier and Relation Recognizer & Event Compositor. Each component includes single or multiple Maximum Entropy models trained by gold annotation data. The inputs of the system are source data and analyzed data. The former is raw text with entity annotation, and the latter is tokenized, POS tagged and parsed data of the raw text.[1]

Because the event type is useful to recognize the relation, we perform TC phase before RE phase.

One of important characteristics of bio-event is that one event as well as a protein may participate in another event. Considering this, we designed the system in which the Relation Recognizer be performed through two steps. In the initial step, the systems examines each trigger whether it has the relations with only proteins, and composites events with recognized relations. In the iterative step, it repeatedly examines remained triggers in the same manner. This step allows the system to extract chain-style events, which means that one event participates in another one and the other participates in the former.

To increase the f-score, we tuned a threshold for RE phase which is a binary classification task; deciding whether a given relation candidate is correct one or not. When the output probability of a maximum entropy model is lower than the threshold, we discard a relation candidate.

2.1 Event Trigger Detection

We assume that an event trigger is a single word. In other words, we do not consider the multi-word trigger detection. Because the trigger statistic in the training data showed that about 93% of triggers are single word, we concentrated on the single word trigger detection.

This phase is simply defined as the task that classify whether each token is a trigger or not in a document. It is necessary to select targets to classify among all tokens, because a set of all tokens includes too many negative examples. For this, the following filtering rules are applied to each token. Though these rules filtered out 69.5% of tokens, the trigger recall was 94.8%.

- Filter out tokens whose POS tag is not matched to anything among NN, NNS, VB, VBD, VBG, VBN, VBP, VBZ, JJ and JJR.
- Filter out tokens that are a biomedical named entity.
- Filter out sentences that do not have any proteins.

Proposed features for the binary classification of tokens include both features similar to those used in (Hacioglu, 2004; Tsai et al., 2006; Ahn, 2006) and novel ones. The selected feature set is showed in Table 1.

2.2 Event Type Classification

In TC phase, tokens recognized as trigger are classified into nine pre-defined classes. Although more than a dozen features had been tested, the features except word and lemma features hardly contributed to the performance improvement. The tuned feature set is showed in Table 2.

[1] We used the *GDep* result provided by organizers of the shared task as analyzed data.

Word level features
- Token word
- Token lemma
- Token POS
- POSs of previous two tokens
- Distance, word and POS of the nearest protein
- Positional independence: Whether a noun or a verb is adjacent to the current token
Dependency level features
- Dependency label path of the nearest protein
- The existence of protein in family: This feature is motivated by the study in (Hacioglu, 2004)
- A boolean feature which is true if token's child is a proposition and the chunk of the child include a protein
- A boolean feature which is true if token's child is a protein and its dependency label is OBJ

Table 1: Features for event trigger detection

Features for the event type classification
- Trigger word
- Trigger lemma
- A boolean feature which is true if a protein exists within left and right two words

Table 2: Features for event type classification

Word level features
- Trigger word
- Trigger lemma
- Trigger type (I-1)
- Entity word
- Entity type (I-2)
- Word sequence between T&P (I-1)
- Word distance
- Existence of another trigger between T&P
- The number of triggers of above feature
- Existence of another participant candidate
- The number of participants of above feature
Dependency level features
- Trigger dependency label (I-1)
- Entity dependency label
- Lemma of trigger's head word (I-1)
- POS of trigger's head word
- Lemma of entity's head word (I-1)
- POS of entity's head word
- Lemma of trigger's head word + Lemma of entity's head word
- Right lemma of trigger's head word
- 2nd right lemma of trigger's head word (I-1)
- Right lemma of entity's head word
- 2nd right lemma of entity's head word (I-1)
- Dependency path between T&P
- Dependency distance between T&P
- Direct descendant: a participant candidate is a direct descendant of a given trigger

Table 3: Features for relation recognition between a trigger and a participant (T&P)

We found that TC phase showed relatively high precision and recall with simple lexical features in the experiment. However, it was quite difficult to find additional features that could improve the performance.

2.3 Relation Recognition and Event Composition

In the last phase, the system examines each trigger whether it has relations with participant candidates, and composites events with the extracted relations. (A relation consists of one trigger and one participant)

We devised a two-step process, consisting of initial and iterative steps, because a participant candidate can be a protein or an event. In the initial step, the system finds relations between triggers and protein participant candidates. Features are explained in Table 3. Then, it generates one event with one relation for event types that have only one participant. For *Binding* event type, the system combines at most three relations of the same trigger into one event. For *Regulation* event types, we trained a binary classifier to classify participants of a *Regulation* event into *theme* or *cause*. Features for participant type classification is explained in Table 4. Among multiple participants of a *Regulation* event, only two participants having highest probabilities for *theme* and *cause* constitute one event.

In the iterative step, the system finds relations between triggers and event participant candidates that were extracted in the previous step, and generates events in the same manner. The system performs iterative steps three times to find chain events.

Features are basically common in the initial (I-1) step and the iterative (I-2) step, but some features improve the performance only in one step. In order to represent the difference in Table 3, we indicate (I-1) when a feature is used in the initial step only, and indicate (I-2) when it used in the iterative step only.

Word level features
- Trigger word
- Trigger lemma
- Participant words - event's trigger words if a participant is an event
- Left lemma of a participant
- Right lemma of a participant
- Trigger word + Participant words
- Trigger lemma + Participant lemmas
- Participant lemmas
- Right lemma of a trigger
- 2nd right lemma of a trigger
- Right lemma of a participant
- 2nd left lemma of a participant
Dependency level features
- Dependency path
- Dependency relation to trigger's head
- Dependency relation to participant's head
- POS pattern of common head chunk of a trigger and a participant
- POS pattern of common head chunk of a trigger and a participant + The presence of an object word in dependency path

Table 4: Features of the participant type classifier for *Regulation* events

3 Experimental Result

Table 5 shows the official results of the final test data. After the feature selection, we have performed the experiments with the development data to tune the threshold to be used in RE phase. The work improved the performance slightly. The new threshold discovered by the work was 0.65 rather than the default value, 0.5. However, we found that the tuned threshold was over-fitted to development data. When we tested without any threshold change, the proposed system showed better f-score by reducing the gap between precision and recall. Table 6 shows the performance in this case.

Nevertheless, recall is still quite lower than precision in Table 6. The reason is that many triggers are not detected in TD phase. The recall of the trigger detector was 63% with the development data. Analyzing errors of TD phase, we found that the system missed terms such as *role, prevent* while it easily detected bio-terms such as *phosphorylation, regulation*. It implies that the word feature causes not only high precision but also low recall in TD phase.

Event equality	**recall**	**precision**	**f-score**
Strict	8.99	58.97	15.60
Approximate Span	9.40	61.65	16.31

Table 5: The official results with threshold tuning

Event equality	**recall**	**precision**	**f-score**
Strict	15.46	52.85	23.92
Approximate Span	16.18	55.32	25.04

Table 6: The results without threshold tuning

4 Conclusion

In this paper, we have presented a biomedical event extraction system consisting of trigger detector, event type classifier and two-step participant recognizer. The system uses dependency parsing and predicate argument information as main sources for feature extraction.

For future work, we would like to increase the performance of TD phase by adopting two-step method similar to RE phase. We also will exploit more analyzed data such as phrase structure parsing information to improve the performance.

References

Kadri Hacioglu. 2004. *Semantic Role Labeling Using Dependency Trees*. In *Proceedings of COLING-2004*, Geneva, Switzerland.

Richard Tzong-Han Tsai, Wen-Chi Chou, Yu-Chun Lin, Cheng-Lung Sung, Wei Ku, Ying-shan Su, Ting-Yi Sung and Wen-Lian Hsu. 2006. *BIOSMILE: Adapting Semantic Role Labeling for Biomedical Verbs: An Exponential Model Coupled with Automatically Generated Template Features*. In *Proceedings of BioNLP-2006*.

Mihai Surdeanu, Sanda Harabagiu, John Williams and Paul Aarseth. 2003. *Using Predicate-Argument Structures for Information Extraction*. In *Proceedings of ACL-2003*, Sapporo, Japan.

David Ahn. 2006. *The stages of event extraction*. In *Proceedings of Workshop On Annotating And Reasoning About Time And Events*.

Jin-Dong Kim, Tomoko Ohta, Sampo Pyysalo, Yoshinobu Kano and Jun'ichi Tsujii. 2009. *Overview of BioNLP'09 Shared Task on Event Extraction*. In *Proceedings of Natural Language Processing in Biomedicine (BioNLP) NAACL 2009 Workshop*.