Improving Feature-Based Biomedical Event Extraction System by Integrating Argument Information

Lishuang Li, Yiwen Wang, Degen Huang
School of Computer Science and Technology
Dalian University of Technology
116023 Dalian, China
lilishuang314@163.com yeevanewong@gmail.com
huangdg@dlut.edu.cn

Abstract

We describe a system for extracting biomedical events among genes and proteins from biomedical literature, using the corpus from the BioNLP'13 Shared Task on Event Extraction. The proposed system is characterized by a wide array of features based on dependency parse graphs and additional argument information in the second trigger detection. Based on the Uturku system which is the best one in the BioNLP'09 Shared Task, we improve the performance of biomedical event extraction by reducing illegal events and false positives in the second trigger detection and the second argument detection. On the development set of BioNLP'13, the system achieves an F-score of 50.96% on the primary task. On the test set of BioNLP'13, it achieves an F-score of 47.56% on the primary task obtaining the 5th place in task 1, which is 1.78 percentage points higher than the baseline (following the Uturku system), demonstrating that the proposed method is efficient.

1 Introduction

Extracting knowledge from unstructured text is one of the most important goals of Natural Language Processing and Artificial Intelligence. Resources in the internet are expanding at an exponential speed, especially in the biomedical domain. Due to the astronomical growth of biomedical scientific literature, it is very important and urgent to develop automatic methods for knowledge extraction system.

In the past few years, most researchers in the field of Biomedical Natural Language Processing focused on extracting information with simple structure, such as named entity recognition (NER), protein-protein interactions (PPIs) (Airola et al., 2008; Miwa et al., 2009) and disease-gene association (Chun et al., 2006). While PPIs concern the flat relational schemas with no nested structures, bio-molecular events describe the detailed behavior of bio-molecules, which capture the biomedical phenomena from texts well. The BioNLP'09 shared task (Kim et al., 2009) provides the first entry to bio-event extraction. As described in BioNLP'09, a bio-event consists of a trigger and one or more arguments, where a trigger is a contiguous textual string containing one or more tokens and an argument is a participant (event or protein) with a corresponding type. For example, in the snippet "*interferon regulatory factor 4 gene expression*", the event trigger is "*expression*" which is tagged by the event type "Gene_expression" and the event argument is "*interferon regulatory factor 4*". Notably, bio-events may have arbitrary arguments and even contain other events as arguments, resulting in nested events.

The complex event structure makes this task particularly attractive, drawing initial interest from many researchers. Björne et al.'s (2009) system (referred to hereinafter as Uturku system) was the best pipeline system in BioNLP'09, achieving an F-score of 51.95% on the test data sets. After that, Miwa et al. (2010a, 2010b) compared different parsers and dependency representations on bio-event extraction task and obtained an F-score of 57.79% on development data sets and 56.00% on test data sets with parser ensemble. In contrast to the pipeline system which divided the event process into three stages, triggers detection, arguments detection and post processing, Poon and Vanderwende's (2010) and Riedel et al.'s (2009) joint models combined trigger recognition and argument detection by using a Markov logic network learning approach. After the BioNLP'09, the Genia event task (BioNLP'11 task 1, hereafter) in the BioNLP'11 Shared Task (Kim et al., 2011) introduced a same event extraction task on a new dataset. There were still some pipeline systems applied to Genia task 1, e.g. Björne et al.'s (2011) system and Quirk et al.'s (2011) system. To the best of

Proceedings of the BioNLP Shared Task 2013 Workshop, pages 109–115,
Sofia, Bulgaria, August 9 2013.

our knowledge, Miwa et al.'s (2012) pipeline system incorporating domain adaptation and coreference resolution, is the best biomedical event extraction system on BioNLP'11 task 1 so far.

The Genia event extraction task (BioNLP'13 task 1, hereafter) (Kim et al., 2013) in BioNLP'13 Shared Task is consistent with the Genia task in BioNLP'11 Shared task. Nevertheless, BioNLP'13 task 1 focuses on event extraction from full texts while BioNLP'11 task 1 contains abstracts and full texts. Furthermore, the coreference resolution task separated from event extraction task in BioNLP'11 is integrated to BioNLP'13 task 1, and there are more event types in the BioNLP'13 task 1 than those in BioNLP'11 task 1. The BioNLP'13 shared task contains three parts, the training corpus, the development corpus and the test corpus. The training corpus consists of 10 full texts containing 2792 events. The development corpus for optimizing the parameters involves 10 full texts containing 3184 events, while the test corpus is composed of 14 full texts including 3301 events. To avoid the researchers optimizing parameters on the test corpus, it is not published, and we have the permission to combine the training corpus and the development corpus as training set. However, we extend BioNLP'13 training set by adding the abstracts of training set and development set in BioNLP'11 task 1 rather than merging the development set of BioNLP'13 into the training set.

Our system generally follows the Uturku system reported by Björne et al. (2009), and uses a simple but efficient way to reduce the cascading errors. The Uturku system was a pipeline of trigger detection, argument detection and post-processing. Each of its components was simple to implement by reducing event extraction task into independent classification of triggers and arguments. Moreover, the Uturku system developed rich features and made extensive use of syntactic dependency parse graphs, and the rules in the post-processing step were efficient and simple. However, the stages of the pipeline introduced cascading errors, meaning that the trigger missed in the trigger detection would never be recalled in the following stages. By changing the pipeline and adding argument information in trigger detection, we construct a model for extracting complex events using rich features and achieve better performance than the baseline system implemented according to Björne et al.'s (2009) paper.

2 Our Event Extraction System

Fig.1 shows the overall architecture of the proposed system. Since 97% of all annotated events are fully contained within a single sentence, our system deals with one sentence at a time, which does not incur a large performance penalty but greatly reduces the size and complexity of the machine learning problems (Björne et al., 2009). The system's components are different from those of the Uturku system by adding a second trigger detection component and a second edge detection component (argument detection). Trigger detection component is used to recognize the trigger words that signify the event, and edge detection component is used to identify the arguments that undergo the change. Semantic post-processing component generates events consistent with the restrictions on event argument types and combinations defined in the shared task.

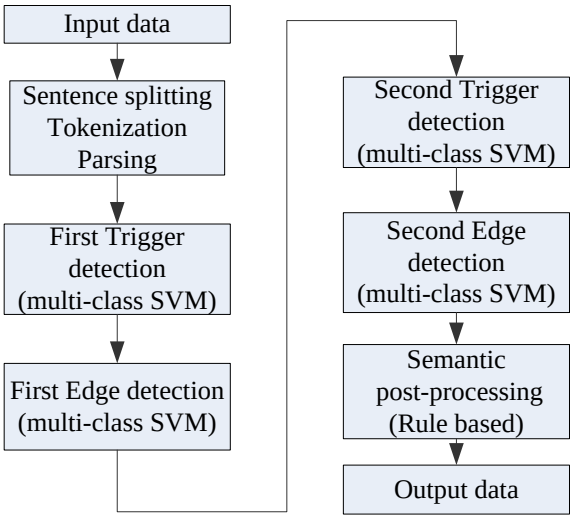

Figure 1. The flow chart of our system.

In the following sections, we present the implementation for these stages in our biomedical event extraction system in detail and evaluate our system on the BioNLP'13 data sets.

2.1 Trigger Detection

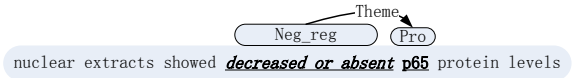

Figure 2. An example of the trigger consisting of two head tokens

Trigger detection assigns each token an event class or a negative class (if the token is not a trigger). The head token is chosen when the real trigger consists of several tokens, which does not

Type	Feature
Primary features	The token
	Part-Of-Speech of the token
	Base form
	The rest part of the token, getting rid of the stem word
Token feature	Token has a capital letter
	Token has a first letter of the sentence
	Token has a number
	Token has a symbol like "-","/","\"
	N-grams (n = 2, 3) of characters
Govern and Dependent feature	Dependency type
	Part-Of-Speech (POS) of the other token
	Combine the POS and the dependency type
	The word form of the other token
Frequency features	Number of named entities in the sentence
	Bag-of-word counts of token texts in the sentence
Shortest path	Token features of the token in the path
	N-grams of dependencies (n =2, 3, 4)
	N-grams of words (base form + POS) (n =2, 3, 4)
	N-grams of consecutive words (base form + POS) representing
	Governor-dependent relationships (n =1, 2, 3)

Table 1: Features for the first trigger detection

Type	Feature
Path feature	The token in the path
	The POS of the token in the path
	The dependency type of edges in the path
	(all these features are combined with direction, length and the entity type)

Table 2: Added feature for the second trigger detection

incur performance penalty with the approximate span matching/approximate recursive matching mode (Kim et al., 2009). Two head tokens may be chosen from one trigger when the trigger consists of two appositives. For example, for the snippets "*decreased or absent p65 protein levels*", both "*decreased"* and "*absent"* are the head token of the trigger "*decreased or absent"*, shown in Fig 2. Rich features are extracted for the first trigger detection, shown in Table 1.

To remove the erroneous events and correct the event type assigned in the first trigger detection, a second trigger detection is added in our system. Thus the second trigger detection is different from the first one. Uturku system shows that the trigger information improves the edge detection because of the constraints on the type of arguments. Naturally, the edge information is helpful for trigger detection with the same reason. As a result, this method can improve the precision of trigger performance.

In order to leverage the argument information, we explore a lot of features of the edges which are the arguments detected in the first edge detection. The edge information concerns the features of the edges attached to the token. In the second trigger detection, we add all the path features between the candidate trigger and arguments attached to the candidate trigger detected in the first edge detection. These features contain the entity information of the argument, the dependency path between the trigger and the argument and so on. Specially, the added features cannot contain any trigger type information obtained in the first trigger detection, or the added features cannot do any help. The reason is that SVM classifier will classify samples only relying on the label feature if it is in the feature set. The added features are shown in Table 2.

Type	Features
N-grams	N-grams of consecutive tokens(n=2,3,4) in the path N-grams of vertex walks
Terminal node feature	Token feature of the terminal nodes The entity type of the terminal nodes Re-normalized confidences of all event class
Frequency feature	The length of the path The number of entities in the sentence
Edges feature in the path	Dependency type of the edges in the path The POS of the tokens in the path The tokens in the path

Table 3: Features for edge detection

2.2 Edge Detection

Similar to the trigger detector, the edge detector is based on a multi-class SVM classifier. An edge is from a trigger to a trigger or from a trigger to a protein. The edge detector classifies each candidate edge as a theme, a cause, or a negative denoting the absence of an edge between the two nodes in the given direction. The features in edge detection are shown in Table 3. As the trigger information is helpful in edge detection, the terminal node feature contains it. Additionally, the first edge detection is completely the same as the second one, that is, they share the same features and machine learning strategy.

2.3 Semantic Post-processing

After the trigger detection and edge detection, the biomedical event cannot be produced directly. Some simple events may be attached with several proteins, and complex events may form circles. We develop a custom rule-based method to generate events that are consistent with the restrictions on event argument types and combinations defined in the shared task. For details, Björne et al.'s (2009) paper can be referred to.

3 Tools and Component Combination

We use the support vector machine (SVM) multi-class classifier (Crammer and Singer (2002), Tsochantaridis et al. (2004)) in the trigger detection and edge detection. Besides, the dependency parser used in our system is McClosky-Charniak domain-adapted parser (McClosky and Charniak (2008)) and the dependency parse was provided in the share task[1]. To optimize the precision-recall trade-off, we introduce β that decreases the classifier confidence score given to the negative trigger class as formula (1) as the Uturku system does (2009).

$$score = score-(1-\beta)*abs(score) \quad (1)$$

where abs(*score*) means the absolute value of score and $\beta \in [0,1]$.

4 Evaluations and Discussion

4.1 Evaluations

Firstly, our system is evaluated on the development set. Table 4 compares the performance between our system and the baseline. The baseline is implemented based on Björne et al.'s (2009) paper. Compared to baseline, the precision of our system is 6.08 percentage points higher while the recall increases 0.91 percentage points. From Table 4 we can see that our system is 2.85 F-score higher than the baseline system.

	Recall	Precision	F-score
Baseline	43.15	54.37	48.12
Ours	44.06	60.45	50.97

Table 4: Performance comparison on the development set using approximate span and recursive matching

Secondly, the performance of our system is evaluated on the test data set with online evaluation[2]. Table 5 shows the results for the baseline and the proposed system with argument information to evaluate the importance of argument information. Integrating argument information, our system archives 1.78% F-score improvement. Compared to the baseline, the performance for complex events is very encouraging with about 7.5 percentage points improvement in the Phosphorylation events, 1.77 percentage points improvement in the regulation events, 2.91 percen-

[1] http://2013.bionlp-st.org/supporting-resources

[2] http://bionlp-st.dbcls.jp/GE/2013/eval-test/

Event type	#	Our system	Baseline
		R/P/F-score	R/P/F-score
Gene_expression	619	77.54/82.76/80.07	79.48/78.10/78.78
Transcription	101	49.50/65.79/56.50	53.47/62.79/57.75
Protein_catabolism	14	78.57/55.00/64.71	78.57/45.83/57.89
Localization	99	35.35/89.74/50.72	38.38/84.44/52.78
=[SIMPLE ALL]=	833	69.15/80.56/74.42	71.43/75.80/73.55
Binding	333	40.84/44.16/42.43	42.64/44.65/43.63
Protein_modification	1	0.00/0.00/0.00	0.00/0.00/0.00
Phosphorylation	160	75.00/77.42/76.19	69.38/68.10/68.73
Ubiquitination	30	0.00/0.00/0.00	0.00/0.00/0.00
Acetylation	0	0.00/0.00/0.00	0.00/0.00/0.00
Deacetylation	0	0.00/0.00/0.00	0.00/0.00/0.00
=[PROT-MOD ALL]=	191	62.83/77.42/69.36	58.12/68.10/62.71
Regulation	288	15.28/42.72/22.51	14.58/35.90/20.74
Positive_regulation	1130	29.20/44.47/35.26	26.11/42.51/32.35
Negative_regulation	526	26.81/41.47/32.56	25.10/35.11/29.27
=[REGULATION ALL]=	1944	26.49/43.46/32.92	24.13/39.51/29.96
==[EVENT TOTAL]==	3301	40.81/57.00/47.56	39.90/53.69/45.78

Table 5: Approximate span matching/approximate recursive matching on test data set.

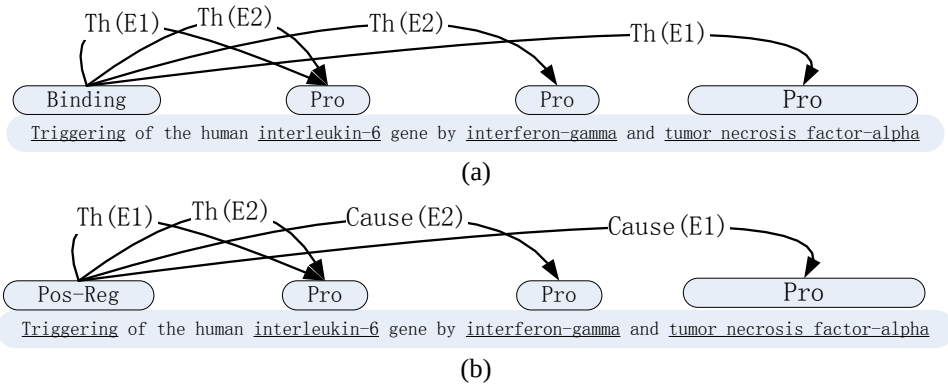

Figure 3: (a) A result of a fragment using the first trigger detection. (b) A result of a fragment using the second trigger detection.

tage points improvement in the positive regulation events and 3.29 percentage points increase in the negative regulation events, but not much loss in other events. As a consequence, the total F-score of our system is 47.56%, 1.78 percentage points higher than the baseline system and obtains the 5th place in BioNLP'13 task 1.

4.2 Discussion

Our system achieves better performance than the baseline thanks to the second trigger detection. The second trigger detection improves the performance of event extraction in two ways. Firstly, the triggers that cannot form events are directly deleted, and therefore the corresponding erroneous events are deleted. Secondly, since the erroneous triggers are deleted or the triggers recognized in the first trigger detection are given the right types in the second trigger detection, the corresponding arguments are reconstructed to form right events. Fig.3 shows an example. In the first trigger detection, the trigger "*triggering*" is recognized as the illegal type of "*binding*" so that "*interferon-gamma*" and "*tumor necrosis factor-alpha*" are illegally detected as theme arguments of "*triggering*", resulting in erroneous events. However, in the second trigger detection,

"*triggering*" is correctly revised as the type of positive regulation, so the arguments are reconstructed, which makes the positive regulation events (E1 and E2) right. As a result, the precision of event detection increases as well as the recall.

The proposed method is an efficient way to reduce cascading errors in pipeline system. Moreover, Riedel and McCallum (2011) proposed a dual decomposition-based model, another efficient method to get around cascading errors. Following Riedel et al.'s (2011) paper, we implement a dual decomposition-based system using the same features in our system. Table 6 shows the performance comparison on the development set of BioNLP'09 between our system and dual decomposition-based system. The comparison indicates that the proposed method is comparable to the stat-of-the-art systems.

	Recall	**Precision**	**F-score**
Dual Decomposition	50.08	63.66	56.06
Ours	53.88	59.67	56.63

Table 6: Performance comparison on the development set of BioNLP'09 using approximate span and recursive matching based on different methods

5 Conclusions

We proposed a simple but effective method to improve event extraction by boosting the trigger detection. The added edge information in the second trigger detection improves the performance of trigger detection. Features from the dependency parse graphs are the main features we use for event extraction.

The future work includes: the first trigger detection should classify a token into three classes: simple event type, complex event type and none event type; discovering some more helpful edge features in the second trigger detection; solving coreference problem with coreference resolution approach. Besides, the dual decomposition-based method will be improved and further compared with the pipeline system.

Acknowledgments

This work is supported by grant from the National Natural Science Foundation of China (no. 61173101, 61173100).

References

Antti Airola, Sampo Pyysalo, Jari Björne, Tapio Pahikkala, Filip Ginter, and Tapio Salakoski. 2008. All-paths graph kernel for protein-protein interaction extraction with evaluation of cross-corpus learning. *BMC Bioinformatics*, 9(Suppl 11):S2.

Chris Quirk, Pallavi Choudhury, Michael Gamon, and Lucy Vanderwend. 2011. MSR-NLP Entry in BioNLP Shared Task 2011. In *Proceedings of the BioNLP 2011 Workshop Companion Volume for Shared Task*, Portland, Oregon, June. Association for Computational Linguistics.

David McClosky and Eugene Charniak. 2008. Self-training for biomedical parsing. In *Proceedings of ACL-08: HLT, Short Papers*, pages 101–104. Association for Computational Linguistics.

Hoifung Poon, Lucy Vanderwende. 2010. Joint Inference for Knowledge Extraction from Biomedical Literature. In *Proceedings of the North American Chapter of the Association for Computational Linguistics-Human Language Technologies 2010 conference*.

Hong-Woo Chun, Yoshimasa Tsuruoka, Jin-Dong Kim, Rie Shiba, Naoki Nagata, Teruyoshi Hishiki, and Jun'ichi Tsujii. 2006. Extraction of gene-disease relations from medline using domain dictionaries and machine learning. In *Proceedings of the Pacific Symposium on Biocomputing (PSB'06)*, pages 4–15.

Ioannis Tsochantaridis, Thomas Hofmann, Thorsten Joachims, and Yasemin Altun. 2004. Support vector machine learning for interdependent and structured output spaces. *In Proceedings of the Twenty-first International Conference on Machine Learning (ICML'04)*, pages 104–111. ACM.

Jari Björne, Juho Heimonen, Filip Ginter, Antti Airola, Tapio Pahikkala, and Tapio Salakoski. 2009. Extracting complex biological events with rich graph-based feature sets. In *Proceedings of the BioNLP 2009 Workshop Companion Volume for Shared Task*, pages 10–18, Boulder, Colorado, June. Association for Computational Linguistics.

Jari Björne and Tapio Salakoski. 2011. Generalizing biomedical event extraction. In *Proceedings of the BioNLP 2011 Workshop Companion Volume for Shared Task*, Portland, Oregon, June. Association for Computational Linguistics.

Jin-Dong Kim, Tomoko Ohta, Sampo Pyysalo, Yoshinobu Kano, and Junichi Tsujii. 2009. Overview of BioNLP'09 Shared Task on event extraction. In *Proceedings of the NAACL-HLT 2009 Workshop on Natural Language Processing in Biomedicine (BioNLP'09)*. ACL.

Jin-Dong Kim, Sampo Pyysalo, Tomoko Ohta, Robert Bossy, and Jun'ichi Tsujii. 2011. Overview of Bi-

oNLP Shared Task 2011. In *Proceedings of the BioNLP 2011 Workshop Companion Volume for Shared Task*, Portland, Oregon, June. Association for Computational Linguistics.

Jin-Dong Kim, Yue Wang and Yamamoto Yasunori. 2013. The Genia Event Extraction Shared Task, 2013 Edition - Overview. In *Proceedings of the BioNLP Shared Task 2013 Workshop*, Sofia, Bulgaria, Aug. Association for Computational Linguistics.

Koby Crammer and Yoram Singer. 2002. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2:265–292.

Makoto Miwa, Rune Sætre, Yusuke Miyao, and-Jun'ichi Tsujii. 2009. A rich feature vector for protein–protein interaction extraction from multiple corpora. In *EMNLP'09: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 121–130, Morristown, NJ, USA. Association for Computational Linguistics.

Makoto Miwa, Sampo Pyysalo, Tadayoshi Hara, and Jun'ichi Tsujii. 2010a . A comparative study of syntactic parsers for event extraction. In *Proceedings of BioNLP'10* p. 37–45.

Makoto Miwa, Sampo Pyysalo, Tadayoshi Hara, and Jun'ichi Tsujii. 2010b. Evaluating dependency representation for event extraction. In *Proceedings of the 23rd International Conference on Computational Linguistics, COLING '10,* Association for Computational Linguistics, 2010; p. 779–787.

Makoto Miwa, Paul Thompson, and Sophia Ananiadou. 2012. Boosting automatic event extraction from the literature using domain adaptation and coreference resolution. *Bioinformatics*.

Sebastian Riedel, Hong-Woo Chun, Toshihisa Takagi,and Jun'ichi Tsujii. 2009. A Markov logic approach to bio-molecular event extraction. In *BioNLP'09: Proceedings of the Workshop on BioNLP*, pages 41-49, Morristown, NJ, USA. Association for Computational Linguistics.

Sebastian Riedel and Andrew McCallum. 2011. Robust Biomedical Event Extraction with Dual Decomposition and Minimal Domain Adaptation. In *Proceedings of the BioNLP 2011 Workshop Companion Volume for Shared Task*, Portland, Oregon, June. Association for Computational Linguistics.