Extracting Biomedical Event with Dual Decomposition Integrating Word Embeddings

Lishuang Li, Shanshan Liu, Meiyue Qin, Yiwen Wang, and Degen Huang

Abstract—Extracting biomedical event from literatures has attracted much attention recently. By now, most of the state-of-the-art systems have been based on pipelines which suffer from cascading errors, and the words encoded by one-hot are unable to represent the semantic information. Joint inference with dual decomposition and novel word embeddings are adopted to address the two problems respectively in this work. Word embeddings are learnt from large scale unlabeled texts and integrated as an unsupervised feature into other rich features based on dependency parse graphs to detect triggers and arguments. The proposed system consists of four components: trigger detector, argument detector, jointly inference with dual decomposition and rule-based semantic post-processing, and outperforms the state-of-the-art systems. On the development set of BioNLP'09, the F-score is 59.77% on the primary task, which is 0.96% higher than the best system. On the test set of BioNLP'11, the F-score is 56.09% and 0.89% higher than the best published result that do not adopt additional techniques. On the test set of BioNLP'13, the F-score reaches 53.19% which is 2.22% higher than the best result.

Index Terms—biomedical event extraction; dual decomposition; word embeddings; natural language processing

1 INTRODUCTION

WITH the development of the Internet, a vast and ever-expanding body of natural language text is becoming increasingly difficult to leverage. This is particularly true in the domain of life science, where biomedical articles are increasing exponentially. We need to automatically extract interested and structured information from biomedical text, which is known as biomedical text mining.

In past years, the major focus of biomedical text mining has been named entity recognition (NER), which identifies entities such as genes, proteins, drugs, and binary relations between such entities. In recent years, text mining researchers pay more attention to complex information extraction, such as biomedical event extraction, with the appearance of applicable NER systems. Biomedical event extraction concerns the detailed behavior of bio-molecules and shows the event information in a structured form, which can represent more detailed and complex relations. The behaviors of bio-molecules mainly include expression, transcription, catabolism, phosphorylation, localization, binding and regulation of genes or proteins. As shown in Fig. 1(c), the text describes regulation and phosphorylation of protein "4E-BP1". Take the event: phosphorylation of "4E-BP1" for an example, the trigger and argument of this event are "phosphorylation" and "4E-BP1" respectively.

There are three shared tasks (ST) related with GENIA event (GE) extraction, BioNLP'09 [1], BioNLP'11 [2] and BioNLP'13 [3]. There are 24, 15 and 12 teams participating in the core task, GE in the three shared tasks respectively. Although these tasks attracted many experts and scholars and many methods were proposed, the task is still a challenge.

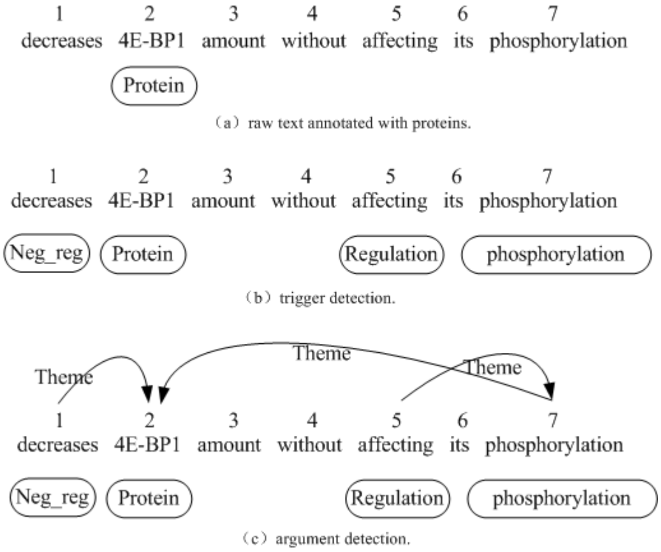


Fig. 1. An example of pipeline-based event extraction method. The number by abovethe word is the index of the word in a sentence.

Several state-of-the-art systems are pipeline-based, including trigger recognition, argument detection and post-processing, as shown in Fig. 1. Björne et al.'s system TEES [4] regarded trigger and argument detection as classification problems. They adopted support vector machine

- L. Li is with the College of Computer Science and Technology, Dalian University of Technology, Dalian 116024, Liaoning, China. E-mail: lilishuang314@ 163.com.
- S. Liu is with the College of Computer Science and Technology, Dalian University of Technology, Dalian 116024, Liaoning, China. E-mail: liushanqust@126.com.
- M. Qin with the College of Computer Science and Technology, Dalian University of Technology, Dalian 116024, Liaoning, China. E-mail: qinmessiy@163.com
- Y. Wang is with the College of Computer Science and Technology, Dalian University of Technology, Dalian 116024, Liaoning, China. E-mail: yeevanewong@gmail.com.
- D. Huang is with the College of Computer Science and Technology, Dalian University of Technology, Dalian 116024, Liaoning, China. E-mail: huangdg@dlut.edu.cn.

xxxx-xxxx/0x/$xx.00 © 200x IEEE Published by the IEEE Computer Society

(SVM) to classify triggers and arguments. Rule-based post-processing was conducted to remove improper combination of arguments. They achieved an F-score of 51.95% on GE09 test data and ranked the first. Björne et al.'s system [5], a pipeline-based event extraction system, included three steps: trigger detection, argument detection and unmerging. It was the most similar to [4] developed for GE09. The major difference was the replacement of the rule-based unmerging component with an SVM based one. It achieved an F-score of 53.30% and ranked the third on the test data of GE11. Hakala et al.'s system EVEX [6] in GE13 extracted events first applying the unmodified TEES system [4] and subsequently re-ranked its output with SVMrank. The re-ranker assigned a numerical score to event produced by TEES, and all events below a certain threshold score were removed, where the threshold was learned adopting linear SVM regressor. Their method achieved the best with an F-score of 50.97% on GE13 task. Björne et al.'s system TEES-2.1 [7] added a module named Automated Annotation Scheme Learning into TEES, a machine-learning based tool for extracting event. Their system ranked the second with an F-score of 50.74% on the GE13 task.

The event extraction systems mentioned above were based on pipelines and several adopted external resources, including trigger recognition and argument detection. The pipeline-based systems suffer from cascading errors. If a trigger is not detected in trigger recognition step, their argument will never be detected and finally the event will be lost. This phenomenon has an adverse effect on the performance of the system. In recent years, joint models have been proposed. Riedel et al. [8] and Poon et al. [9] adopted Markov logic network and manually made predicate logic joint statements to extract triggers and arguments simultaneously. They could get the F-scores as high as 43.1% and 50.0% on BioNLP'09 test set respectively. Although Markov logic network could avoid cascading errors, the performance did not exceed the pipeline systems due to the complex structure of biomedical events and the shortcoming that Markov logic network could not make good use of a large number of features. Riedel et al.'s system UMass [10] adopted passive-aggressive (PA) online learning algorithm to predict the confidence of triggers and arguments, and then extracted the events with the highest confidence and some constraints using dual decomposition. They achieved the best F-scores of 57.4% and 55.2% on the test set of GE09 and GE11 respectively without adopting any additional technologies.

In previous works, the way to digitalize features is one-hot encoding. The main problem of this method is that it is unable to represent the semantic information. Recently, word embeddings, a vector related with a word, are used in several NLP problems, such as named entity recognition (NER), chunking, and make a contribution to the improvement. Tang et al. [11] explored the effect of word embeddings on biomedical NER. Turian et al. [12] discussed the impact on several tasks, including NER and chunking. In part-of-speech tagging task, Fonseca et. al [13] explored the influences of kinds of word embeddings learnt from different models, including neural language model(NLM), skip-grams and hyper-space analogue to language(HAL). These researches indicate word embeddings are conducive to different natural language processing tasks.

Considering the two problems of cascading errors and semantic information absence mentioned, we propose this method: adopting dual decomposition and rich features integrating word embeddings to detect event jointly. Dual decomposition has been used in other natural language processing task, such as named entity recognition [14] and Chinese discharge summaries [15] and has been proved beneficial to improve performance. In addition, word embeddings, a type of word representation, are firstly used in event extraction to our best knowledge. The main strengths of our work are: 1) Rich features based on dependency parse. 2) Word embeddings, novel word features which can represent word semantically and syntactically. 3) Dual decomposition which can extract event jointly using inference and alleviate cascading errors. In this work, we use a dual decomposition method and adopte rich features based on dependency parse. Furthermore, an unsupervised word feature, called word embeddings is integrated into the rich features. In dual decomposition, the Passive-aggressive (PA) online algorithm [16] is adopted to allocate confidence to triggers and arguments.

The remaining part of this paper is organized as follows: the related work is described in Section 2. Our proposed method is described in Section 3. Experimental results and analysis are illustrated in Section 4. Finally, conclusions are drawn in Section 5.

2 Preliminary Algorithms

2.1 Online Passive-aggressive Algorithms

Passive-aggressive (PA) online algorithm [16] is an online algorithm based on perception. The main idea of the algorithm is the maximum classification margin adopted in SVM. It updates the classifier using the instance greedily and predicts the instance correctly with the maximum margin and remains the new classifier as close as possible to the current one.

The pseudo code of Passive-aggressive online algorithm is shown in Fig. 2, y_t is a trigger or argument type and x_t is feature vector. Y is the set of y_t . *N* is the number of iterations. *w* is the weight needed to be learned. Multi-class PA can assign a score to each class of an instance, which can provide additive confidence for dual decomposition. Note that all parameters in the algorithm are optimized on developing sets. In order to improve the robustness of a classifier and reduce the number of possible combinations, several outstanding classifiers' models after optimized on the parameter C are selected and the mean of selected models is adopted. In our work, the trigger class and argument class with the highest scores are the predicted results when using online algorithms.The interested readers can refer to [16] for more details.

```
Input: parameter $C > 0$, $N$
Initialize weights: $w_1 = (0,...,0)$
for $t=1,2,...,N$
  1. Receive example: $x_t \in R^n$
  2. Predict: $\hat{y}_t = \arg\max_{y \in Y} (w_t \cdot \Phi(x_t, y))$
        where $\Phi(x_t, y)$ is features related with class y
  3. Receive correct class: $y_t \in Y$
  4. Loss function :
  $\ell_t = \max\{0,\ 1 - w_t \cdot \Phi(x_t, y_t) + w_t \cdot \Phi(x_t, \hat{y}_t)\}$
  5. update:
     set:
     $\tau_t = \min\left\{C,\ \frac{\ell_t}{\|x_t\|^2}\right\}$
     update: $w_{t+1} = w_t + \tau_t y_t x_t$
```

Fig. 2. Passive-aggressive online algorithm.

2.2 Dual Decomposition

Dual decomposition is a combined optimization method and widely used to solve complicate problems. It usually decomposes a hard problem into two simple problems with constraints. The process of extracting events can be decomposed into simple problems with constraints 1) Outgoing constraints (O): There is at least one Theme argument for a trigger; Only regulation event are allowed to have Cause arguments; a None trigger must have no arguments, 2) Incoming constraints(I): arguments must be proteins or another trigger. For a nested event, the trigger of its argument (another event) must participate in a complete event. The process is described in detail in [10].

To introduce this algorithm briefly, several binary variables are defined. $e_{i,t} = 1$ represents that the *i*th token in a sentence is a trigger with type t . $a_{i,j,r} = 1$ represents that the *j*th token is the argument of the trigger i with the role of r . In Fig. 1, $e_{1,Neg\_reg} = 1$, $a_{1,2,Theme} = 1$. For a combination of a trigger and an argument, its score can be defined as (1).

$$s(e,a) \overset{def}{=} \sum_{e_{i,t}=1} s_T(i,t) + \sum_{a_{i,j,r}=1} s_R(i,j,r) \qquad (1)$$

where $s(e,a) \in O \wedge s(e,a) \in I$, $\sum_{e_{i,t}=1} s_T(i,t)$ means the score of the trigger i with type t and $\sum_{a_{i,j,r}=1} s_R(i,j,r)$ represents the score of the jth token being the argument of the trigger i with the role of r .

The scoring function is defined as (2).

$$s_m(y; x, w) =< w, f(y,x) > \qquad (2)$$

where w is the weight used in PA, $f(y,x)$ is the feature vector.

The event with the highest $s(e,a)$ is the final result extracted by dual decomposition.

2.3 Word Embeddings

A distributed representation, also known as word embeddings, is dense, low dimensional, and real-valued. Word embeddings are typically induced using neural language models, which use neural networks as the underlying predictive model. There are several word embeddings, such as Collobert and Weston embeddings(C&W) [17], HLBL embeddings [18] and Word2Vec [19, 20].

Considered the time and hardware requirements in different distributed representation methods, Word2Vec, developed by [19, 20], is adopted in our work. Word2Vec has two models: CBOW and Skip-gram. The Skip-gram model extended from n-gram model is used and shown in Fig. 3. It aims to optimize the classification of a word by other words in the same sentence within a certain range. This tool can generate a dense, low-dimensional, and real-valued vector, which may capture the syntactic and semantic information in each dimension. This information cannot be obtained from words encoded by one-hot.

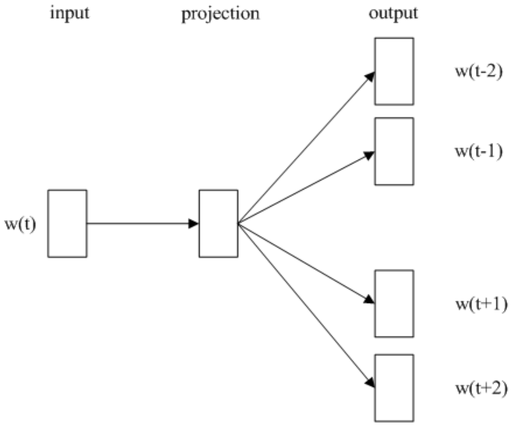


Fig. 3. The Skip-gram architecture.

To train better word embeddings, a large scale of unlabeled texts are required. The way we train word embeddings is as follows: First, abstract texts are downloaded from the public database, PubMed, with the size of about 5.6G. Then, all abstracts are split into sentences and tokenized into tokens. Finally, all tokenized sentences are sent into the tool Word2Vec to get the vectors. The parameters, window sizes and dimensions, are set 7 and 400 respectively when training the vectors finally in our work.

3 Method Description

The overall framework of the proposed method is shown in Fig. 4. The system mainly has four components: preprocessing, trigger and argument prediction, dual decomposition and rule-based post-processing.

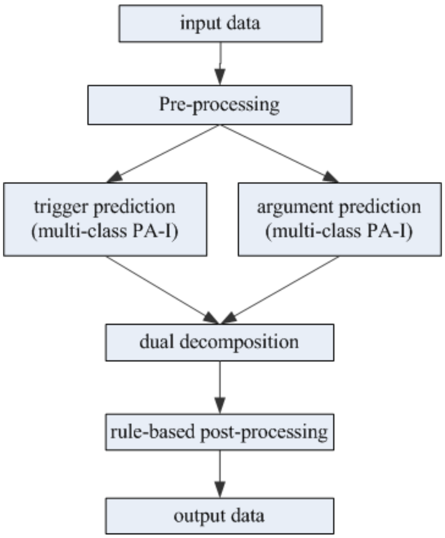

input data
Pre-processing
trigger prediction (multi-class PA-I)
argument prediction (multi-class PA-I)
dual decomposition
rule-based post-processing
output data

Fig. 4. The framework of the proposed method.

3.1 Pre-processing

The pre-processing mainly includes sentence splitting, tokenization, and syntactic and dependency parsing. The raw texts are in paragraphs and are split into sentences using Genia Sentence Splitter. Supporting resources provided by BioNLP'11 are adopted to correct common errors and tokenize all sentences. The tokenized sentences are parsed with McClosky parser [21] and Enju parser [22].

3.2 Trigger Prediction

Trigger prediction is to assign confidence scores indicating the credibility of a token to be different types of trigger. It is expected to allocate a highest score to the correct type of a token while the lowest to the irrelevant types. It is depicted as token classification task in this work. To train a good classifier, a wide array of valid features is extracted from text after pre-processing, such as token itself, n-gram syntactic and dependency parsing information, and word embeddings. The flow chart of trigger prediction is shown in Fig. 5.

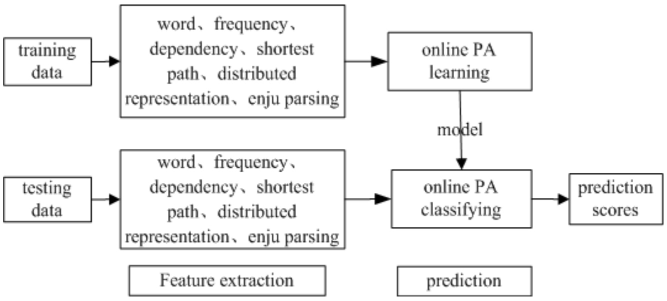

training data
word、frequency、dependency、shortest path、distributed representation、enju parsing
online PA learning
model
testing data
word、frequency、dependency、shortest path、distributed representation、enju parsing
online PA classifying
prediction scores
Feature extraction
prediction

Fig. 5. The flow chart of trigger prediction.

In this work, five kinds of features are mainly used, token, frequency, dependency chains, shortest path and word embeddings. The dependency paths parsed by McClosky-Charniak parser [21] and Enju parser [22] are added into the features.

Token features include current token text, POS, stem, binary tests for presence of uppercase, digital or special characters, bigrams and trigrams of the token. Dependency context is of great importance for trigger detection, so we extract token features of candidate triggers in dependency context and linear context besides candidate triggers themselves.

- Token text includes current token and the tokens within a window of three tokens before and after the target tokens.
- POS includes the POS of the current token and the tokens within a window of three tokens. The POS is tagged with McClosky-Charniak parser [21]. The POS distribution of triggers in BioNLP'09 training set is shown in Table 1.
- Stem consists of the stem of the current token, obtained by Porter stemmer [23]. This feature can alleviate the effect of morphological changes, such as "involvement" and "involves".
- Binary features include binary tests for presence of uppercase, digital or special characters. Some words with a negative class may contain digitals or capital letters. Some triggers contain special characters, such as "up-regulation", "co-transfected".
- Bigrams and trigrams consist of two or three continuous characters in current token. For example, for the token "binding", its trigrams are "bin", "ind", "ndi", "din", and "ing".

TABLE 1

THE PROPORTION OF TRIGGERS POS IN BIONLP'09 TRAINING SET

POS	PERCENTAGE(%)	POS	PERCENTAGE(%)
NN	49.9	VB	4.3
VBN	13.6	NNS	3.8
JJ	8.1	VBG	3.4
VBD	6.5	VBP	2.8
VBZ	5.9	OTHERS	1.7

Frequency features are defined as the number of named entities in the sentence and the context of a candidate trigger, and the frequency of words in bag-of-words. It is obvious that the more entities in a sentence there are, the more likely triggers exist in the sentence. For the frequency of words in bag-of-words, we take this sentence for an example, "The p53 paradox in the pathogenesis of tumor progression.", the frequency of words in its bag-of-words are "*the:2*", "*p53:1*", "*paradox:1*", "*in:1*", "*pathogenesis:1*", "*of:1*", "*tumor:1*", "*progression:1*", "*.:1*" and "*PROTEIN:1*". Here, the protein names are all replaced with "PROTEIN".

Dependency chains up to depth of three are constructed. At each depth, both token features and dependency types are included, as well as the sequence of dependency types in the chain. Because of the limitation of linear context, if the linear window size is small, some important information related with candidate triggers cannot be considered and therefore dependency information is added.

- *Token features of nodes in dependency chains* include POS of the token, the token and whether the node is protein or not. These features are added with position information (the distance from proteins) in dependency chains.

- *Dependency types in dependency chains* are also added with position information, sequence of dependency types and direction. An example of dependency parsing is shown as Fig. 6. For the token "inhibits", its dependency chains features are: "*1_binding*", "*1_NN*", "*1_dobj*", "*1_dobj_NN*", "*1_dobj_binding*", "*1_Phosphorylation*", "*1_NN*", "*1_nsubj*", "*1_nsubj_NN*", "*1_nsubj_Phosphorylation*", etc.

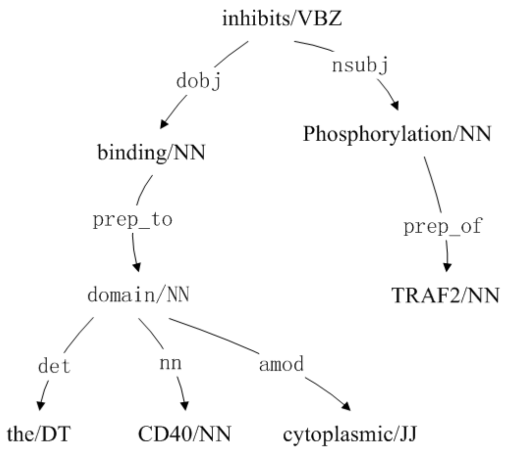


Fig. 6. An example of dependency parsing.

Shortest path is a directed path including the path information between the candidate trigger and its nearest protein. Concretely, the path information includes *n*-grams (*n*=2, 3, 4) of the edges in the shortest dependency path between candidate trigger and the nearest protein, and the combinations of the entity types in the shortest path. There is an important connection between the candidate trigger and the nearest protein. Furthermore, the nearest protein is more likely to be the argument of trigger. As shown in Fig. 6, for the candidate word "inhibits", its nearest protein is "CD40", and the path between them is "*inhibits-dobj->binding-prep_to->domain-nn->CD40*". The *n*-gram feature is represented as an *n*-tuple of vertexes and edges in the shortest path, where the vertexes are extracted from the word window "*inhibits, binding, domain, CD40*", and the edges are obtained from the string window "dobj, prep_to, nn". Besides, the combinations of entity types can be gained from "NN, NN, NN".

Word embeddings refer to the vectors of the current token. The dimension of the vectors is decided by experiments.

3.3 Argument Prediction

Similar with trigger prediction, argument prediction is also treated as a classification task and online PA multiclass classifier is adopted. Each instance will be assigned three scores representing the probabilities of a candidate argument to be Theme, Cause or Negative. Theme and Cause are the semantic role of arguments.

The flow chart of argument detection is similar with trigger detection (shown as Fig. 5) just with different features in the step of feature extraction. The features in this step mainly include: N-grams, individual component features, semantic node features, frequency, shortest path, word embeddings and dependency edges.

According to [4], the distances among event and named entity in the shortest dependency path are shorter than those in linear order of the sentence. Therefore the features are almost constructed on the base of the shortest dependency path. The two terminal ends of the shortest dependency path are the semantic heads of triggers or named entities, which can be obtained using rules.

N-grams features are obtained from dependency paths of candidate arguments. N-grams combine up to 4 continuous tokens and dependency relations. Each token and the dependency types on the both sides, and the dependency type and its two attached tokens are constructed. In Fig. 6, for "TRAF2", its N-grams features are: "*nsubj-Phosphorylation-prep_of*", "*inhibits-nsubj-Phosphorylation*", "*Phosphorylation-preo_of-TRAF2*" and "*prep_of-TRAF2*".

Individual component features are defined for the edges and tokens in a path according to their attributes and positions, where positions mean the inner or the end of the path. Edge attributes are combined with their direction relative to the path.

Semantic node features are constructed by combining the attributes of the two nodes of candidate arguments. These features concatenate the types of nodes and their categories.

Word embeddings are built with the vectors of the current candidate trigger and argument.

Adjacent dependency edges include POS, dependency type, the stem of word and entity type, together with dependency direction in the dependency paths adjacent with the current candidate argument.

Frequency features include the length of the shortest path (the longer the length is, the less likely the argument is), and the number of named entities in the current sentence.

Shortest path includes *n*-grams (*n*=2, 3, 4) of the edges in the shortest dependency path between candidate argument and the nearest protein, the dependency type and semantic node features in the dependency path.

3.4 Rule-based Semantic Post-processing

We adjust the results which may not meet the definition of event in the task using rules after trigger and argument detection. The rules are similar with those proposed by [4]. So we do not introduce them in detail.

4 Results and Analysis

4.1 Corpus and Evaluation

All experiments are conducted on the corpora provided by BioNLP'09, BioNLP'11 and BioNLP'13. And parameters are optimized on development set. The evaluation criteria "Approximate Span/Approximate Recursive" and P(recision)/R(ecall)/F(-score) are adopted. Due to the inaccessibility of the evaluation interface on testing set of BioNLP'09 and BioNLP'11, we just show the performance on development set. The evaluation metric P/R/F is defined as below (3), where *TP*, *FP* and *FN* are short for

True Positives, False Positives and False Negatives respectively.

$$P=\frac{TP}{TP+FP},\ R=\frac{TP}{TP+FN},\ F\text{-}score=\frac{2*P*R}{P+R} \tag{3}$$

4.2 Results of Trigger and Argument Prediction Integrating Word Embeddings

Five groups of experiments are conducted on the development set of BioNLP'09 to select the optimal dimension of the vectors. The dimension of the vectors is set to 50, 100, 200 and 400 respectively to compare the influence of word embeddings on trigger and argument prediction. The results are shown in Table 2 and Table 3, and our baseline is that using all features except word embeddings. BaselineWE50, BaselineWE100, BaselineWE200 and BaselineWE400 mean the dimensions of word embeddings are 50, 100, 200 and 400 respectively when word embeddings are integrated. The type with the highest score is the final predict result.

TABLE 2
THE INFLUENCE OF DIMENSIONS ON TRIGGER PREDICTION

Features	Gold(match)	Answer(match)	P/R/F(%)
Baseline	1335(891)	1194(891)	74.62/66.74/70.46
BaselineWE50	1335(915)	1218(915)	75.12/68.54/71.68
BaselineWE100	1335(920)	1230(920)	74.80/68.91/71.73
BaselineWE200	1335(920)	1221(920)	75.35/68.91/71.99
BaselineWE400	1335(959)	1310(956)	71.57/71.84/71.70

TABLE 3
THE INFLUENCE OF DIMENSIONS ON ARGUMENT PREDICTION

Features	Gold(match)	Answer(match)	P/R/F(%)
Baseline	1954(1236)	1767(1236)	69.95/63.25/66.43
BaselineWE50	1954(1230)	1727(1230)	71.22/62.95/66.83
BaselineWE100	1954(1261)	1801(1261)	70.02/64.53/67.16
BaselineWE200	1954(1248)	1742(1248)	71.64/63.87/67.53
BaselineWE400	1954(1249)	1745(1249)	71.58/63.92/**67.53**

From Table 2 and Table 3, we can see all of the F-scores of trigger and argument prediction using word embeddings are improved compared with Baseline. The F-score improves with the increase of dimension except the dimension of 400 on trigger prediction. The F-scores are improved by 1.22~1.53% and 0.4~1.1% with the variance of the dimension of the vectors, which illustrates that the syntactic and semantic information carried by word embeddings has significantly increases the performance.

4.3 Result of Event Extraction Integrating Word Embeddings

To verify the effect of the word embeddings feature, the results with/without word embeddings on BioNLP'09 development set are shown in Table 4 (without dual decomposition). Word embeddings improve Binding and Regulation events significantly by 1.48% and 2.81% respectively, though the F-score of Simple events is decreased slightly by 0.12% caused by the degradation of Transcription and Protein_catabolism event. Finally word embeddings improve the F-score by 1.45% for event extraction.

TABLE 4
RESULTS WITH/WITHOUT WORD EMBEDDINGS ON EVENT DETECTION ON DEVELOPMENT SET OF BIONLP'09

Event Class	Without WE	With WE
	P/R/F(%)	P/R/F(%)
Gene_expression	83.54/77.25/80.27	83.64/77.81/**80.62**
Transcription	85.25/63.41/**72.73**	86.21/60.98/71.43
Protein_catabolism	87.50/74.47/**80.46**	81.40/74.47/77.78
Localization	97.56/75.47/85.11	95.35/77.36/**85.42**
=[SVT-TOTAL]=	85.77/74.60/**79.80**	85.28/74.78/79.68
Binding	63.01/37.10/46.70	60.74/39.92/**48.18**
==[EVT-TOTAL]==	80.51/63.07/70.73	79.14/64.06/**70.81**
Regulation	61.29/33.73/43.51	64.42/39.64/**49.08**
Positive_regulation	59.56/39.38/47.41	60.87/40.84/**48.88**
Negative_regulation	60.87/35.71/45.02	59.57/42.86/**49.85**
==[REG-TOTAL]==	60.06/37.68/46.31	61.15/41.04/**49.12**
==[ALL-TOTAL]==	70.41/49.13/57.88	70.10/51.43/**59.33**

4.4 Results of Event Extraction Using Dual Decomposition

We compare the performance using these two methods: PA online and dual decomposition. To simplify the tables, we just list the performance on Simple, Binding, Modification, Regulation event, and Task 1 here and after.

In Table 5, the F-scores on all classes of events are improved by 0.34%, 2.18% and 0.39% respectively. Dual decomposition enhances the final performance of event extraction by 0.44% especially with the improvement of Binding event (increased by 2.18% F-score) on the development set of BioNLP'09. As shown in Table 6, on the development set of BioNLP'11, the performances of all classes of events are increased especially Binding event increased by 1.88%. The F-score is finally improved by 0.61% with the respective improvement on Simple (0.45%), Binding (1.88%) and Regulation event (0.41%). On the test set of BioNLP'13 shown in Table 7, the F-scores of events are increased with different degrees except slight degradation of Binding event with 0.32%. The final F-score on Task 1 is 0.28% improved. Thus dual decomposition increases the performance on the three corpora.

From Table 4 and Table 5, we can see word embeddings and dual decomposition improve the F-scores by 1.45% and 0.44% respectively. Thus word embeddings are valuable word features (obtained from unsupervised method) for event extraction. Except Binding event in BioNLP'13, the performance on other events is improved based on dual decomposition. It can be summarized from Table 5, Table 6 and Table 7 that dual decomposition and rich features integrating word embeddings can improve the F-scores on the three corpora by 0.28%~0.61%.

TABLE 5

RESULTS USING ONLINE MULTI-CLASS PA AND DUAL DECOMPOSITION ON DEVELOPMENT SET OF BIONLP'09

Event Class	Online multi-class PA	Dual decomposition
	P/R/F (%)	P/R/F (%)
Simple	85.28/74.78/79.68	83.59/76.74/80.02
Binding	60.74/39.92/48.18	61.27/42.74/50.36
Regulation	61.15/41.04/49.12	57.30/43.58/49.51
Task 1	70.10/51.43/59.33	67.18/53.83/59.77

TABLE 6

RESULTS USING ONLINE MULTI-CLASS PA AND DUAL DECOMPOSITION ON DEVELOPMENT SET OF BIONLP'11

Event Class	Online multi-class PA	Dual decomposition
	P/R/F (%)	P/R/F (%)
Simple	85.95/70.76/77.62	83.30/73.47/**78.07**
Binding	61.51/39.41/48.04	60.23/42.63/**49.92**
Regulation	53.80/37.00/43.85	51.46/39.10/**44.44**
Task 1	66.98/48.81/56.47	64.40/51.25/**57.08**

TABLE 7

RESULTS USING ONLINE MULTI-CLASS PA AND DUAL DECOMPOSITION ON TEST SET OF BIONLP'13

Event Class	Online multi-class PA	Dual decomposition
	P/R/F (%)	P/R/F (%)
Simple	74.91/84.32/79.34	76.11/83.31/**79.55**
Binding	45.95/46.36/**46.15**	46.25/45.43/45.83
Modification	66.49/83.55/74.05	68.06/81.25/**74.07**
Regulation	31.53/50.62/38.86	34.21/47.81/**39.88**
Task 1	45.96/62.35/52.91	47.96/59.71/**53.19**

4.5 Comparison with Other Systems

We compare our method with others on three corpora: the development set of BioNLP'09 and the test sets of BioNLP'11 and BioNlP'13. We achieve the best performance on the three corpora as shown in Table 8, 9 and 10 except compared with EventMine [24] which integrated domain adaption and co-reference resolution in Table 9. Due to the inaccessibility of online evaluation of BioNLP'09 test set, we make the comparisons on the development set of BioNLP'09 briefly.

From Table 8, our F-score on the development set of BioNLP'09 is 59.77%. The F-scores are improved by 0.96% and 1.07% respectively than EventMine [24] and UMass [10]. The improvement of Regulation event contributes to the final improvement with the similar performance on Simple and Binding event compared with EventMine. And the improvement of Simple and Binding events contributes to the improvement compared with UMass.

TABLE 8

COMPARISON WITH OTHER SYSTEMS ON THE DEVELOPMENT SET OF BIONLP'09

Event Class	Ours	EventMine	UMass
	P/R/F (%)	P/R/F (%)	F (%)
Simple	83.59/76.74/80.02	80.16	78.4
Binding	61.27/42.74/50.36	50.52	48.0
Regulation	57.30/43.58/49.51	47.48	49.1
Task 1	67.18/53.83/59.77	58.81	58.7

From Table 9, we can see our F-score on BioNLP'11 test set is 56.09% and outperforms the state-of-art systems which did not adopt additional technologies such as co-reference resolution. The performance of EventMine [24] exceeded ours, but it incorporated co-reference resolution and domain adaption into event extraction and did not report the performance before adopting the two technologies. Our F-score is improved by 0.89% than UMass which is the best system on BioNLP'11 test set by now without adopting additional technologies. Our result is superior to UTurku system[25](improved by 2.79% F-score) which extended their BioNLP'09 Shared Task winning Turku Event Extraction System with replacement of the rule-based unmerging component based on SVM and ranked the first in the shared task of BioNLP'11 at that time.

TABLE 9

COMPARISON WITH OTHER SYSTEMS ON THE TEST SET OF BIONLP'11

Event Class	Ours	EventMine	UMass	UTurku
	P/R/F (%)	F (%)	F (%)	F (%)
Simple	84.07/69.05/75.82	-	73.5	-
Binding	58.02/46.44/51.58	-	48.8	-
Regulation	48.77/40.35/44.16	-	43.8	-
Task 1	62.16/51.11/56.09	57.98	55.2	53.3

From Table 10, we achieve an F-score of 53.19% with our proposed method on the test of BioNLP'13. The great improvements on all kinds of events contribute to the final improvement. The results show that dual decomposition integrating word embeddings and rich features is beneficial for event extraction and achieves 2.22% and 2.45% improvement respectively than the systems ranked the first and second on the test of BioNLP'13 (EVEX and TEES-2.1). EVEX [6] re-ranked the events extracted by the unmodified TEES-2.1 [7] using the large-scale text mining resource EVEX and the tool of SVMrank, and achieved 0.23% improvement than TEES-2.1 [7]. TEES-2.1 [7] modified Turku Event Extraction System (TEES) and added the module of automated annotation scheme learning. The features used in TEES-2.1 followed those in TEES. The two systems were based on pipelines and did not adopt additional technologies. Therefore our joint method incorporating word embeddings and rich features achieves the best and shows its strengths compared with pipeline systems.

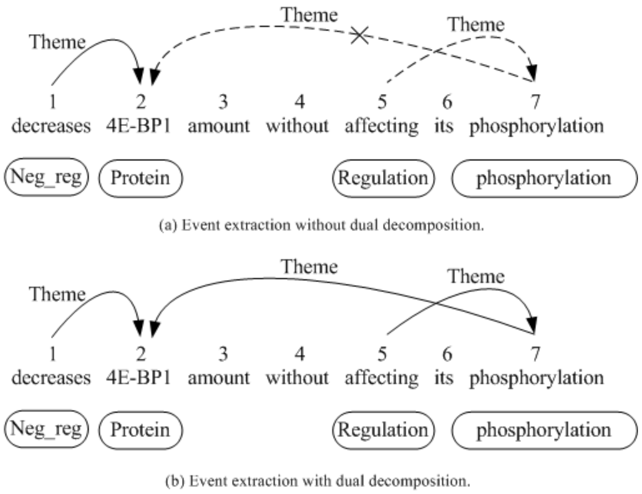
TABLE 10

COMPARISON WITH OTHER SYSTEMS ON THE TEST SET OF BIONLP'13

Event Class	Ours P/R/F (%)	EVEX F (%)	TEES-2.1 F (%)
Simple	83.31/76.11/79.55	76.59	76.82
Binding	45.43/46.25/45.83	42.88	43.32
Modification	81.25/68.06/74.07	65.37	66.49
Regulation	47.81/34.21/39.88	38.41	38.05
Task 1	59.71/47.96/53.19	50.97	50.74

4.6 Analisys

From all above experimental results, our method improves the final performance and precedes other excellent systems which do not integrate additional technologies on existing comparable corpora. The reasons may be 1) The rich features are the solid foundation, such as token features, syntactic and dependency features, the shortest path. 2) Word embeddings, which can learn much deeper syntactic and semantic information from the large set of out-of-domain data obtained through unsupervised learning and adopted innovatively in event extraction, lead to the vectors of words with common semantics are close to each other, and thus improve trigger and argument detection significantly (shown in Table 2 and Table 3). For example, for the two words, "diminished" and "reduced", they have little common features directly in morphology, but the similarity between their word embeddings measured by cosine similarity is up to 0.897. 3) Dual decomposition, which avoids or alleviates cascading errors. Take the fragment of "… decreases 4E-BP1 amount without affecting its phosphorylation ... " from development set as an example shown in Fig. 7.



(a) Event extraction without dual decomposition.

(b) Event extraction with dual decomposition.

Fig. 7. An example to show efficiency of dual decomposition. Solid lines represent identified events; dotted lines with a cross mean missing trigger or argument; dotted lines mean unrecognized events.

Fig. 7(a) is the extracted event without dual decomposition. The three triggers "decreases", "affecting" and "phosphorylation" are classified into three event classes "Negative_regulation", "Regulation" and "phosphorylation" respectively. The protein "4E-BP1" is detected as the argument with semantic role "Theme" of the trigger "decreases", and the trigger "phosphorylation" is detected as the argument with semantic role "Theme" of the trigger "affecting". But the Theme argument "4E-BP1" of the trigger "phosphorylation" is lost. According to the definition of nested event, the Regulation event is not detected.

Fig. 7(b) is the extracted event with dual decomposition. By the experiment tracked, the scores assigned by online PA to the trigger are Phosphorylation: 0.853763 and Negative: 0.704227 respectively. The protein "4E-BP1" is the Theme argument with 0.836427 and Negative with 0.92632 related with the trigger "phosphorylation". According to dual decomposition algorithm, "4E-BP1" is classified as the Theme argument of "phosphorylation" with higher reliability. The Regulation event is detected correctly accordant with the gold event structure of the fragment.

In order to inspect the impact of different combination of trigger and argument scores, the experiments based on nonlinear combinations are also conducted. The experimental results from Table 11 show that the impact of the way of combination is weak. Therefore we choose the simplest linear combination.

TABLE 11

THE RESULTS BASED ON LINEAR AND NONLINEAR COMBINATION ON BIONLP'11 TEST SET

The way of combination		P	R	F
linear combination	$a+b$	62.16%	51.11%	56.09%
nonlinear combination	$a*b$	66.73%	47.61%	55.57%
	$\log a+b$	66.77%	47.61%	55.59%
	$a+\log b$	66.76%	47.68%	55.63%
	a^2+b^2	66.76%	47.59%	55.57%

a and *b* represent the scores of trigger and argument respectively.

In a word, word embeddings is verified significant to event extraction and the performance of event extraction is further improved by integrating the rich features and word embeddings into dual decomposition.

5 CONCLUSION

The proposed method improves the performance of event extraction, outperforming most of published works. First, rich features are the solid foundation. Second, word embeddings play an important role which implies a lot of useful information, including syntactic and semantic. Using word embedding, the performances in the step of trigger and argument prediction are improved, and thus the F-scores on event extraction are improved. Finally, dual decomposition alleviates cascading errors inherent in the pipeline systems and detects more events. By integrating the rich features and word embeddings into dual decomposition, our system outperforms the state-of-the-art systems.

Despite the great efforts, the extraction of complex event is still a challenge. In the future, we will integrate co-reference resolution and domain adaption into the extraction of event. Further researches on how to adopt probabilistic, such as Bayesian approach, and unsupervised or semi-supervised methods to event extraction should be continued.

ACKNOWLEDGMENT

The authors gratefully acknowledge the financial support provided by the National Natural Science Foundation of China under No. 61173101, 61173100.

REFERENCES

[1] J.-D. Kim, T. Ohta, S. Pyysalo, Y. Kano, and J. Tsujii, "Overview of BioNLP'09 shared task on event extraction," *Proc. of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task(pp. 1-9)*. Association for Computational Linguistics, 2009.

[2] J.-D. Kim, Y. Wang, T. Takagi, and A. Yonezawa, "Overview of genia event task in bionlp shared task 2011," *Proc. of the BioNLP Shared Task 2011 Workshop* (2011), pp. 7–15, 2011.

[3] J.-D. Kim, Y. Wang, and Y. Yasunori, "The Genia Event Extraction Shared Task, 2013 Edition-Overview," *ACL 2013*, 8.

[4] J. Björne, J. Heimonen, F. Ginter, A. Airola, T. Pahikkala, and T. Salakoski, "Extracting complex biological events with rich graph-based feature sets," *Proc. of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task* (2009), pp. 10–18.

[5] J. Björne, and T. Salakoski, "Generalizing Biomedical Event Extraction", *Proc. Of BioNLP Shared Task 2011 Workshop*, pp. 183-191, June 2011, Association for Computational Linguistics.

[6] K. Hakala, S.V. Landeghem, T. Salakoski, Y.V. Peer, and F. Ginter, "EVEX in ST'13: Application of a large-scale text mining resource to event extraction and network construction," *Proc. of the BioNLP Shared Task 2013 Workshop*, pp. 26-34, Aug. 2013, Association for Computational Linguistics..

[7] J. Björne, and T. Salakoski, "TEES 2.1: Automated annotation scheme learning in the BioNLP 2013 Shared Task," *Proc. of the BioNLP Shared Task 2013 Workshop*, pp. 16–25, Aug. 2013, Association for Computational Linguistics.

[8] S. Riedel, H.-W.Chun, T. Takagi, and J. Tsujii, "A markov logic approach to bio-molecular event extraction," *Proc. of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task* (2009), pp. 41–49, June 2009, Association for Computational Linguistics.

[9] H. Poon, and L. Vanderwende, "Joint inference for knowledge extraction from biomedical literature," *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics* (2010), pp. 813–821.

[10] S. Riedel, and A. McCallum, "Fast and robust joint models for biomedical event extraction," *Proc. of the Conference on Empirical Methods in Natural Language Processing (2011)*, pp. 1–12, July 2011, Association for Computational Linguistics.

[11] B. Tang, H. Cao, X. Wang, Q. Chen, and H. Xu, "Evaluating word representation features in biomedical named entity recognition tasks," *BioMed research international*, 2014.

[12] J. Turian, L. Ratinov, and Y. Bengio, "Word representations: a simple and general method for semi-supervised learning," *Proc. of the 48th Annual Meeting of the Association for Computational Linguistics*, pp. 384–394,2010.

[13] E. R. Fonseca, J. L. G. Rosa, and S. M. Aluísio, "Evaluating word embeddings and a revised corpus for part-of-speech tagging in Portuguese,"*Journal of the Brazilian Computer Society*, pp.1-14, 2015.

[14] H. L. Chieu, and L. N. Teow, "Combining local and non-local information with dual decomposition for named entity recognition from text," *In Information Fusion (FUSION), 2012 15th International Conference on*, pp. 231-238, 2012.

[15] M. Wang, W. Che, and C. D Manning, "Joint Word Alignment and Bilingual Named Entity Recognition Using Dual Decomposition," *ACL (1)* ,pp. 1073-1082, 2013.

[16] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer, "Online passive-aggressive algorithms," J. of Machine Learning Research., pp. 551–585, 2006.

[17] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *J. of Machine Learning Research*, pp. 2493–2537, 2011.

[18] A. Mnih, and G.E. Hinton, "A Scalable Hierarchical Distributed Language Model," *NIPS*, pp. 1081–1088, 2008.

[19] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality.,"*Advances in Neural Information Processing Systems*, pp. 3111–3119,2013.

[20] T. Mikolov, W. Yih, and G. Zweig, "Linguistic regularities in continuous space word representations," *Proc. of NAACL-HLT* (2013), pp. 746–751, 2013.

[21] D. McClosky, and E. Charniak, "Self-training for biomedical parsing," *Proc. of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, pp. 101–104, 2008.

[22] Y. Miyao, K. Sagae, R. Sætre, T. Matsuzaki, and J. Tsujii, "Evaluating contributions of natural language parsers to protein--protein interaction extraction," *Bioinformatics*, vol. 25, pp.394–400,2009.

[23] M.F. Porter, "An algorithm for suffix stripping," *Program: electronic library and information systems,* vol.14, pp. 130–137,1980.

[24] M. Miwa, P. Thompson, and S. Ananiadou, "Boosting automatic event extraction from the literature using domain adaptation and coreference resolution," Bioinformatics, pp.1759-1765,2012.

[25] J. Björne, and T.Salakoski, "Generalizing biomedical event extraction, " *Proc. of the BioNLP Shared Task 2011 Workshop*, pp. 183-191,2011, Association for Computational Linguistics.

Lishuang Li received her BSc degree from Dalian University of Technology in 1989, the MSc degree and PhD degree from Dalian University of Technology in 1992 and 2013 respectively. She is currently a professor in the School of Computer Science and Technology at Dalian University of Technology. She has published more than 60 research papers in various journals and conferences. Her research interests include text mining, natural language processing and machine translation. In recent years, she has focused on text mining for biomedical literatures and information extraction from huge biomedical resources. Her research projects are funded by the NSFC.

Shanshan Liu received her BSc degree from Qingdao University of Science and Technology in 2012. She is an MSc candidate in the School of Computer Science and Technology at Dalian University of technology. Her research interests include text mining for biomedical literatures and information extraction from huge biomedical resources.

Meiyue Qin received his BSc degree from Inner Mongolia University in 2014. He is an MSc candidate in the School of Computer Science and Technology at Dalian University of

technology. His research interests include text mining for biomedical literatures and information extraction from huge biomedical resources.

Yiwen Wang received his BSc degree from Dalian University of Technology in 2011. He is an MSc candidate in the School of Computer Science and Technology at Dalian University of technology. His research interests include text mining for biomedical literatures and information extraction from huge biomedical resources.

Degen Huang received his MSc degree from Dalian University of Technology in 1988 and his PhD degree from Dalian University of Technology in 2004. He is currently a professor in the School of Computer Science and Technology at Dalian University of Technology. His research interests include machine translation, text mining, natural language processing and machine learning. In recent years, his research projects are funded by the NSFC.