IEICE TRANS. INF. & SYST., VOL.E102–D, NO.9 SEPTEMBER 2019

PAPER

Multi-Level Attention Based BLSTM Neural Network for Biomedical Event Extraction

Xinyu HE[†a)], **Lishuang LI**[†b)], **Xingchen SONG**[†c)], **Degen HUANG**[†d)], ***Nonmembers***, ***and*** **Fuji REN**[††], ***Member***

SUMMARY Biomedical event extraction is an important and challenging task in Information Extraction, which plays a key role for medicine research and disease prevention. Most of the existing event detection methods are based on shallow machine learning methods which mainly rely on domain knowledge and elaborately designed features. Another challenge is that some crucial information as well as the interactions among words or arguments may be ignored since most works treat words and sentences equally. Therefore, we employ a Bidirectional Long Short Term Memory (BLSTM) neural network for event extraction, which can skip handcrafted complex feature extraction. Furthermore, we propose a multi-level attention mechanism, including word level attention which determines the importance of words in a sentence, and the sentence level attention which determines the importance of relevant arguments. Finally, we train dependency word embeddings and add sentence vectors to enrich semantic information. The experimental results show that our model achieves an F-score of 59.61% on the commonly used dataset (MLEE) of biomedical event extraction, which outperforms other state-of-the-art methods.

key words: *event extraction, trigger detection, argument detection, BLSTM neural network, multi-level attention*

1. Introduction

With the rapid spread of Internet, the biomedical literature is expanding at an exponential speed, which has made it harder than ever for scientists to research, manage, and extract knowledge from unstructured text in their research field. To tackle these problems, biomedical information extraction techniques are rapidly developing. As one of the important information extraction areas, biomedical event extraction aims to extract more fine-grained and complex biomedical relations at molecular level, such as biological molecules, cells and tissues [1], which provides inspiration and basis for the diagnosis, treatment, new drug research and development of diseases. Several evaluation tasks have been held in recent years to allow researchers to develop and compare their approaches for biomedical event extraction.

A biomedical event comprises one event trigger and one or more arguments. The event triggers are usually verbs or gerunds which trigger the occurrence of a biomedical event. The arguments are usually biomedical entities or other events that are the participants in a biomedical event. For example, in the sentence fragment "Gi protein (pertussis toxin), prevented induction of 1L-10 production by Gp41 in monocytes" (Fig. 1), which describes the inhibition of "pertussis toxin" to "monocyte", three biomedical events are involved as follows. The first event is a Gene_expression type event E1, including a trigger word "production" and a Theme type argument "1L-10"; the second event is a Regulation type event E2, including a trigger word "induction", a Theme type argument E1, and a Cause type argument "Gp41"; the last event is a Negative_regulation type event E3, including a trigger word "prevented" and a Theme type argument E2. Event E1 is a simple event. Event E2 and E3 are complex events. The structures of the three events are as follows:

Event E1 (Type: Gene_expression, Trigger: production, Theme: 1L-10);

Event E2 (Type: Positive_regulation, Trigger: induction, Theme: Event E1, Cause: Gp41);

Event E3 (Type: Negative_regulation, Trigger: prevented, Theme: Event E2).

Most of the state-of-the-art biomedical event extraction systems are pipeline-based, including three subprocesses: trigger identification, argument detection and post-processing. In the pipeline process, trigger identification and argument detection are usually regarded as classification problems. Particularly speaking, argument detection belongs to complex relation classification. In previous works, most methods of event detection are based on shallow machine learning. Pyysalo et al. [1] and Zhou et al. [2] utilized support vector machine (SVM) [3] to classify triggers and arguments with handcrafted features. Zhou et al. [4] integrated domain knowledge in their architecture.

To reduce the cost of exacting artificial features, deep neural networks have been widely used in NLP tasks. Wang et al. [5] and Nie et al. [6] employed a neural network architecture to identify triggers. Wang et al. [7] employed Convolutional Neural Network (CNN) to extract biomedical events. The above advanced methods have their notable advantages for event extraction. However, there are still

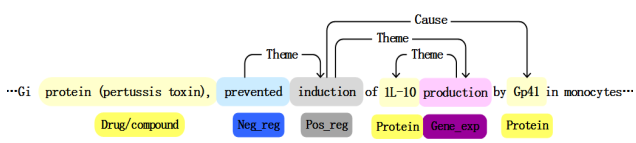

Fig. 1 An example of biomedical event in a fragment of text

Manuscript received August 3, 2018.
Manuscript revised November 15, 2018.
Manuscript publicized April 26, 2019.

[†]The authors are with the Dalian University of Technology, Dalian, China.
[††]The author is with the University of Tokushima, Tokushima-shi, 770–8501 Japan.
a) E-mail: hexinyu@mail.dlut.edu.cn
b) E-mail: lilishuang314@163.com (Corresponding author)
c) E-mail: dut-songxingchen@qq.com
d) E-mail: huangdg@dlut.edu.cn; (Corresponding author)

DOI: 10.1587/transinf.2018EDP7268

some aspects which can be improved. Firstly, most of the mentioned methods rely on hand-crafted features, which are time consuming, and may lead to poor generalization ability. Secondly, the existing methods utilize pre-trained word embeddings, without dependency information and sentence level feature representation which are important for biomedical event extraction task. Last but not least, the approaches mentioned above take all words as equally important and the most crucial semantic information may be ignored.

To solve these problems, we construct a deep learning model via the bidirectional LSTM (BLSTM) to extract biomedical events without additional artificial features. The contributions of this work lie in three-fold:

(1) We train the dependency-based word embeddings with large scale corpus, which contains important dependency information for event extraction.

(2) Based on the pre-trained word embeddings, we supplement fine-tuned word embeddings with a training process to enrich the input information, and calculate sentence vectors to get the global sentence feature representation.

(3) We propose a multi-level attention mechanism for event extraction. The word level attention can enhance the weights of critical words which have decisive effect on trigger and argument detection. The sentence level attention determines the importance of relevant arguments and enhances the interactions among them which can improve the performance of biomedical event extraction significantly.

Based on the above, our method achieves a higher F-score on biomedical event extraction task compared to the state-of-the-art systems, which demonstrates that the proposed method is effective for biomedical event extraction.

2. Related Work

Trigger and argument detection are regarded as multi-class classification tasks in most of the state-of-the-art event extraction systems. On the MLEE corpus, Pyysalo et al. [1] employed a SVM-based approach to extract events. They designed salient lexical and local context features manually, including whether a word has a capital letter or a number and so on. Zhou et al. [2] presented a semi-supervised learning framework to identify the trigger based on hidden topics. In this framework, the hidden topics embedded in the sentences are used for describing the distance. Zhou et al. [4] learned biomedical domain knowledge from a large text corpus built from Medline and embedded it into word features. The above methods rely on the hand-crafted features which are time consuming. Also, different features are needed to tailor for different task, thus not making them generalizable.

Wang et al. [5] employed a neural network architecture to learn significant feature representation based on dependency relation tree, and dynamically adjusted the embeddings while training for adapting to the trigger classification task. Nie et al. [6] proposed a word EANNP architecture to conduct event identification. Wang et al. [7] employed task-based features represented in a distributed way as the input of CNN models to train deep learning models. Although these methods can effectively alleviate the problem of manually extracting features, however, most of them rely on local sentence representation features only within a window, which may be insufficient for event extraction. In addition, the above methods treated all words as equally important, the most crucial semantic information might be ignored. Therefore, we propose a BLSTM-based method to extract events, integrating multi-level attention mechanism in our model, furthermore, the sentence vectors are utilized to capture the global information of the sentences.

3. Methods

In this paper, we propose a BLSTM neural network integrating multi-level attention mechanism and dependency-based word embeddings to extract biomedical events. As shown in Fig. 2, there are four parts in the event extraction, which are the input representation of data, trigger identification, argument detection and post-processing. Firstly, we train dependency-based word embeddings by word2vecf [8] as the input of BLSTMs. To enrich the input information, we add the sentence vector as additional input. Then, the BLSTMs integrating the multi-level attention mechanism are applied in the trigger identification and argument detection. In the trigger identification stage, the word level attention is employed. In the stage of argument detection, we propose a multi-level attention mechanism, which integrates both word level attention and sentence level attention. Finally, the complete biomedical events are constructed by the machine-learning based post-processing.

3.1 Input Representation of Data

3.1.1 Dependency-Based Word Embeddings

In recent years, word embeddings have been widely used in NLP tasks. Miwa et al. [9] has validated the effectiveness of dependency information for biomedical event extraction. In this paper, we train dependency-based word embeddings as feature representation, which can yield more focused embeddings, capturing more functional and less topical similarity.

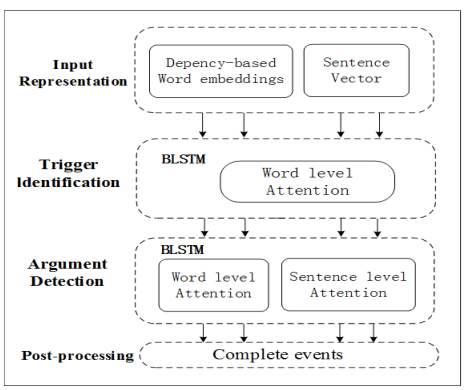

Fig. 2 Our framework of event extraction

In our research, we firstly download 5.7G PubMed abstracts, then parse the abstracts with GDep parser [10], which is a dependency parse tool specialized for biomedical texts. Finally, we utilize Word2vecf [8] to train dependency-based word embeddings with the dependency contexts derived in the previous step.

3.1.2 Sentence Vector

The sentence level features may be ignored only using the word level embeddings. In addition, there is a strong association between events appearing in a sentence, which results in the global information of the sentence is critical to biomedical event extraction. Therefore, in our BLSTM architecture, we integrate sentence information as supplementary inputs. The effectiveness of the sentence vectors has been verified in some NLP tasks, such as biological named entity recognition (NER) [11]. With similar approach, we provide two kinds of word embeddings in the whole training process. One is the pre-trained dependency-based word embeddings x_t, which can obtain the potential feature information from large scale unlabeled corpus. The other is fine-tuned word embeddings x'_t, which contains richer information associated with the biomedical events learned by the neural networks. The initial value of x'_t is the same as x_t, however, x'_t is fine-tuned with BLSTM neural network training process. To take advantage of both the word embeddings, we integrate the sentence vectors in our LSTM framework. The memory cell of the LSTM integrating the sentence vector is shown in Fig. 3. As Eq. (1) shown, the sentence vector d_0 is generated by averaging the differences of all the words' two word embeddings in a sentence. Besides, we use reading gate $r_t \in [0,1]^n$ to control what information should be retrained for future time steps.

$$d_0 = \frac{1}{n}\left(\sum_{t=1}^{T}(x'_t - x_t)\right) \tag{1}$$

3.2 BLSTM Integrating Sentence Vectors and Multi-Level Attention

3.2.1 BLSTM Integrating Sentence Vectors

LSTM units are firstly proposed by Hochreiter and Schmidhuber [12] to retain information over long distances in time successfully. A standard architecture of LSTM mainly consists of three units, which are the input, output and forget gates. The LSTM variants are described as follows:

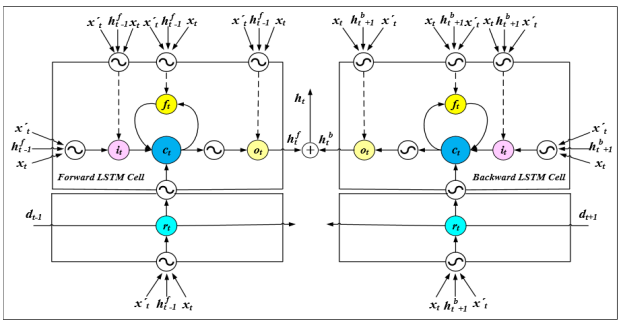

Fig. 3 Memory cell of BLSTM integrating sentence vector

$$i_t = \sigma(x_t \cdot w^i_{xh} + h_{t-1} \cdot w^i_{hh'} + b^i_h) \tag{2}$$

$$f_t = \sigma(x_t \cdot w^f_{xh} + h_{t-1} \cdot w^f_{hh'} + b^f_h) \tag{3}$$

$$o_t = \sigma(x_t \cdot w^o_{xh} + h_{t-1} \cdot w^o_{hh'} + b^o_h) \tag{4}$$

$$\tilde{c}_t = tanh(x_t \cdot w^c_{xh} + h_{t-1} \cdot w^c_{hh'} + b^c_h) \tag{5}$$

$$c_t = i_t \odot \tilde{c}_t + f_t \odot c_{t-1} \tag{6}$$

$$h_t = o_t \odot tanh(c_t), \tag{7}$$

where σ refers to the logistic sigmoid function and $\odot$ denotes the element-wise multiplication. x is the input embeddings at time t, and i, f, o and c are input gate, forget gate, output gate and the proposed values respectively, all of which are the same size as the hidden vector h. w_{xh}, w_{hh} and b_h are the input connections, recurrent connections and bias values respectively. c_t denotes the true cell value at time t.

Bidirectional LSTM (BLSTM) networks can exploit information both from the left and the right contexts [13]. Therefore, BLSTM is used for trigger and argument detection. As shown in Fig. 3, we combine the forward pass output (h^f_t) and the backward pass output (h^b_t) by summation. The output at t moment is shown as Eq. (8). In addition, we employ two kinds of word embeddings in the whole training process, thus our new BLSTM architecture after adding the fine-tuned word embeddings can be described as Eq. (9) to (12). The reading gate can be described as Eq. (13). The sentence information can be calculated as Eq. (14) at t moment. After integrating the sentence vector, the cell value c_t is modified to Eq. (15).

$$h_t = [\overrightarrow{h_t} \oplus \overleftarrow{h_t}] \tag{8}$$

$$i_t = \sigma(x_t \cdot w^i_{xh} + x'_t \cdot w^i_{x'h} + h_{t-1} \cdot w^i_{hh'} + b^i_h) \tag{9}$$

$$f_t = \sigma(x_t \cdot w^f_{xh} + x'_t \cdot w^f_{x'h} + h_{t-1} \cdot w^f_{hh'} + b^f_h) \tag{10}$$

$$o_t = \sigma(x_t \cdot w^o_{xh} + x'_t \cdot w^o_{x'h} + h_{t-1} \cdot w^o_{hh'} + b^o_h) \tag{11}$$

$$\tilde{c}_t = tanh(x_t \cdot w^c_{xh} + x'_t \cdot w^c_{x'h} + h_{t-1} \cdot w^c_{hh'} + b^c_h) \tag{12}$$

$$r_t = \sigma(x_t \cdot w^r_{xh} + x'_t \cdot w^r_{x'h} + h_{t-1} \cdot w^r_{hh'} + b^r_h) \tag{13}$$

$$d_t = r_t \odot d_{t-1} \tag{14}$$

$$c_t = i_t \odot \tilde{c}_t + f_t \odot c_{t-1} + tanh(d_t) \tag{15}$$

3.2.2 Multi-Level Attention

Word Level Attention: Attention-based neural networks have recently demonstrated success in a wide range of tasks ranging from digit classification [14], machine translation [15], to sentence summarization [16]. Inspired by the previous works, we integrate attention mechanism in our BLSTM architecture for the event detection, which helps to filter out the irrelevant noise and find the important units in the input sequence.

Different from most attention mechanisms, we set a random initial weight matrix and let it tune with training process instead of setting a fixed calculation formula for weight. In this way, the common features will be learned by the neural networks automatically, and the corresponding weights of the words with these features will be increased, therefore, the pivotal information can be captured. As shown in Eq. (16), $H \in R^{L \times d_w}$ represents the final BLSTM hidden layer output vector matrix, consisting of output vectors $[h_1, h_2, \cdots, h_L]$, where L is the sentence length, d_w refers to the dimension of the word embeddings. After utilizing the activation function tanh, the attention weights are trained as Eq. (17), where w denotes a trained parameter vector and w^T is its transpose. Then, in Eq. (18), the representation γ of the sentence is formed by a weighted sum of the output vector H, where the dimensions of α and γ are L and d_w respectively. Finally, the ultimate semantic information of the sentence is produced from Eq. (19), where h^* denotes the final sentence representation and the representation of ith word is described as h_i^*.

$$N = tanh(H) \tag{16}$$

$$\alpha = softmax(w^T N) \tag{17}$$

$$\gamma = H\alpha^T \tag{18}$$

$$h^* = tanh(\gamma) \tag{19}$$

Sentence Level Attention: In the event extraction, the argument candidate instances are the pairs comprising the predicted trigger and entity/trigger (nested event). To obtain the context information, we extend the argument candidate instances to sentence fragments which are composed of predicted trigger, entity/trigger (nested event) and the words between them of the raw texts. The word level attention only captures the features from the given argument instances. However, other instances may contain significant semantic information for this instances. Therefore, it is reasonable to look over other relevant instances when determining the type of the current argument instance. We employ the sentence level attention to enhance the interaction among the relevant instances.

In this paper, the argument instances that have the same predicted trigger are believed to be relevant, and they are trained in the same batch. The weights of the important instances will be increased by the attention mechanism mentioned above. The attention can strengthen the interaction of any relevant argument, which can make up for the long distance dependence of LSTM. The relevant instances set is represented as $H^* = \{h_1^*, h_2^*, \cdots, h_M^*\}$, where M denotes the numbers of relevant instances in a batch, h_i^* is the hidden output in the word level attention. After reducing the dimension of h_i^* as Eq. (20) shown, a new vector matrix $H_S^* = \{h_{S_1}^*, h_{S_2}^*, \cdots, h_{S_M}^*\}$ is generated, which represents the sentence feature vector. With similar attention mechanism, replacing H with H_S^* from Eq. (16) to Eq. (19), a new h^* in Eq. (19) denoting the final sentence representation will be calculated, which is used for argument classification by softmax function.

$$h_{S_i}{}^* = \sum_{i=1}^{d_w} \sum_{j=1}^{L} h_i^* / L \tag{20}$$

3.3 Trigger Identification

The goal of trigger identification is to assign each token in a sentence to an event trigger class (19 types in all) or a negative class if it does not belong to a trigger class. In the stage of trigger detection, we firstly treat each token in a sentence as a trigger candidate instance. Then, the hidden state h_i^* of each token is generated by the BLSTM model integrating word level attention. Finally, a softmax classifier is employed to predict label $\hat{y}$ of each trigger candidate. The classifier takes the hidden state h_i^* as input:

$$\hat{p}(y|x) = softmax(Wh_i^* + b) \tag{21}$$

$$\hat{y} = arg \max_y \hat{p}(y|x) \tag{22}$$

After trigger identification, argument detection is needed. Their frameworks are similar, the only difference is that we add the sentence level attention based on the trigger identification model for argument detection, namely multi-level attention. Since the argument detection (Fig. 4) is more complicated, therefore, a detailed argument detection architecture is described in Sect. 3.4, the framework of trigger identification is no longer described here.

3.4 Argument Detection

The goal of argument detection is finding whether there is a relation between an event trigger and an entity, or between an event trigger and another event trigger (nested event). If there is a relation, the relation types, namely argument types (7 types in all), should be predicted.

In the argument detection stage, we firstly generate all potential argument candidate instances based on the predicted triggers and the given entities of the sentences. As mentioned in "Sentence Level Attention", we take the sentence fragments which are composed of predicted trigger, entity/trigger (nested event) and the words between them as argument candidate instances. Then, our argument detection model takes all argument candidate instances as inputs, and outputs argument types (7 types in all) the argument candidate belongs to.

As shown in Fig. 4, the n relevant instances are described as $\{S_1, S_2, \cdots, S_n\}$ which are sentence fragments constructed by the predicted triggers, argument candidates and the words between them, S_i denotes the i-th sentence ($i \in [1,n]$), l_i is the length of S_i, and w_j^i is the j-th word in sentence $S_i (j \in [1, l_i])$. In the embedding layer, the model represents each word (w_j^i) as a vector. After the BLSTM layer, the word level attention layer takes the hidden outputs of BLSTM layer as inputs, and the sentence matrix is generated, where each column vector (S_i) is the semantic representation of the corresponding word. The sentence level attention layer produces final representation for the relevant instances with the same predicted triggers. Finally, the

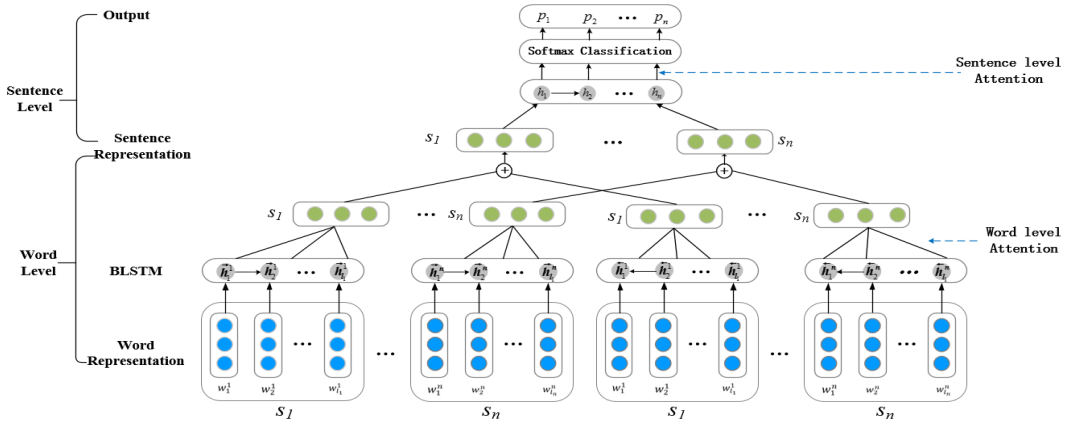

Fig. 4 The framework of argument detection

model classifies the argument candidates as a specific argument type by a softmax classifier.

3.5 Post-Processing

After trigger identification and argument detection, the final complete biomedical events are constituted by post-processing. The post-processing [17] ensures that the events can generate candidate instances correctly. The main methods of post-processing are rule-based and machine learning based. We employ a SVM-based post-processing to learn the valid event combination automatically, which can avoid the cost of handcrafted rules. The SVM classifier automatically learns the legal event structure for each event type by the extracted features, then constitutes event candidates and finally determines their event types. The features extracted in this process mainly include three categories [18]: linear span features, argument combination features and argument content features.

4. Experiments and Results

4.1 Corpus and Evaluation

Our experiments are conducted on the commonly used dataset (MLEE) [1]. The MLEE dataset supports event extraction of all levels of biomedical organization from the molecular level to the whole organism. Specifically, the related event types are divided into four categories (i.e. Anatomical, Molecular, General and Planned), containing 19 pre-defined event categories. The static distribution of the MLEE corpus is shown as Table 1.

On the MLEE corpus, the data is provided in three parts as training, development and test sets. We combine the training and development datasets for training, use development dataset for tuning parameters, and the test dataset for testing. We evaluate the proposed approach with P(recision)/R(ecall)/F(-score). The evaluation metric P/R/F is defined as Eq. (23), where TP, FP and FN are short for True Positives, False Positives and False Negatives respectively.

Table 1 The static distribution of the MLEE corpus

Data	Train	Test	Total
Document	175	87	262
Sentence	1728	880	2608
Event	4471	2206	6677

$$P = \frac{TP}{TP + FP}, R = \frac{TP}{TP + FN}, F\text{-}score = \frac{2 * P * R}{P + R} \quad (23)$$

4.2 Hyper-Parameters Settings and Training Details

Our framework is implemented based on Theano, and the number of BLSTM neural network layer is 2. We use a batch size of 64 in all training. The dimensions of all the word-embeddings employed in the experiments are 200. In addition, we set the dropout [19] rate to 0.5, the number of hidden nodes to 200, and the maximum number of iterations to 100 epochs. We select the Adadelta as the training algorithm. The learning rate is selected as 0.01 from the set {0.01, 0.001, 0.0001}.

4.3 Results of Trigger Prediction

Dependency-based Word Embeddings vs. Skip-gram Model based Word Embeddings: The dependency-based word embeddings can obtain more abundant semantic information, so that the performance of trigger detection will be improved. As shown in Table 2, dependency-based word embeddings (line 2) outperforms the skip-gram word embeddings (line 1) by 2.05% F-score, which illustrates the effectiveness of dependency-based word embeddings.

Effectiveness of Sentence Vectors: We generate the sentence vectors by the following ways:(1) averaging or maximizing the differences (/sum) between the pre-trained and the fine-tuned word embeddings of each word in sentences; (2) averaging or maximizing only the fine-tuned or pre-trained word embeddings. Finally, averaging the differences between the two kinds of word embeddings obtains the best performance, achieving an F-score of 77.96%. As shown in Table 2 (line 4), the F-score of our method is improved by 3.75% significantly, which validates the effective-

ness of the sentence vectors.

Effectiveness of Attention Mechanism: Based on the above experiments, the attention mechanism is integrated. As shown in Table 2 (the last line), the experimental result has been increased to 79.55%, improved by 1.59% F-score. It is worth mentioning that the recall is improved by 2.9% F-score after integrating attention mechanism. The main reason is that the attention mechanism helps to filter out the irrelevant noise and find the important units in the input sequence, so that more event trigger words are recalled.

4.4 Results of Argument Prediction and Event Extraction

Effectiveness of Multi-level Attention Mechanism: The results of argument prediction and event extraction integrating multi-level attention are shown in Table 3. To constitute more biomedical events, we select the results of argument prediction with higher recall instead of higher F-score. That is, Recall is the measure of model performance and we report the results with the best Recall of each model in Table 3. As shown in Table 3, the recall of argument detection with word level attention (line 2) and sentence level attention (line 3) are both improved than the baseline method (line 1). However, the recall is the highest when we combine word level attention with sentence level attention (line 4), namely multi-level attention. Table 3 also give the corresponding results of event extraction. The corresponding F-score of event extraction is also better when the word level attention or sentence level attention is integrated into our model. When we integrate multi-level attention, the performance of event extraction is best, achieving an F-score of 59.61%, which illustrates the effectiveness of multi-level attention.

4.5 Comparisons with Other Methods

Recent researches of event extraction are all based on the MLEE corpus, and there is no published results of argument detection for comparison. Therefore, we compare our experimental results of trigger identification and event extraction with the advanced methods based on MLEE.

Table 2 Performances of different methods on trigger detection

Method	P (%)	R (%)	F (%)
LSTM(Skip-gram)	69.53	71.64	70.57
LSTM(Dependency-based)	73.56	71.70	72.62
BLSTM(Dependency-based)	76.26	72.27	74.21
BLSTM+Sentence Vector	82.81	73.66	77.96
BLSTM+Sentence Vector+Attention	82.79	76.56	79.55

4.5.1 Comparisons of Trigger Identification Performance of Different Methods

Pyysalo et al. [1] utilized support vector machine (SVM) to classify triggers, achieving 75.84% F-score. Zhou et al. [2] identified the trigger based on hidden topics, which obtained 76.89% F-score. Wang et al. [5] employed a neural network architecture to learn better feature representation for trigger identification, achieving a micro F-score of 78.27%. Nie et al. [6] proposed a word embeddings assisted neural network architecture (EANNP) to conduct event identification. Their system achieved an F-score of 77.23%. Zhou et al.'s [4] trigger identification method integrating domain knowledge achieved 78.32% F-score. As shown in Table 4, our method achieves the best performance on the MLEE corpus. Our F-score is 3.71% higher than the baseline method's, 1.23% higher than Zhou et al.'s [2] best performance system, 1.28% higher than Wang et al.'s [5], and 2.22% higher than Nie et al.'s [6], which also employed deep learning methods in their studies.

4.5.2 Comparisons of Event Extraction Performance of Different Methods

From Table 5, our approach achieves 59.61% F-score on MLEE datasets for event extraction, which is 1.3% higher than the best system proposed by Wang et al. [7]. They employed task-based features represented in a distributed way as the input of CNN models to extract biomedical events. Zhou et al. [2] utilized a semi-supervised learning frame work based on hidden topics for biomedical event extraction. Pyysalo et al. [1] employed an SVM-based approach

Table 4 Comparison of the performance of trigger identification

Method	P (%)	R (%)	F (%)
Pyysalo et al. [1]	70.79	81.69	75.84
Zhou et al. [2]	72.17	82.26	76.89
Nie et al. [6]	71.04	84.60	77.23
Wang et al. [5]	73.56	83.62	78.27
Zhou et al. [4]	75.35	81.60	78.32
Proposed	82.79	76.56	**79.55**

Table 5 Comparison of the performance of event extraction

Method	P (%)	R (%)	F (%)
Pyysalo et al. [1]	62.28	49.56	55.20
Zhou et al. [2]	55.76	59.16	57.41
Wang et al. [7]	60.56	56.23	58.31
Proposed	90.24	44.50	**59.61**

Table 3 Performances of different methods on argument detection and event extraction

	Argument Prediction			Event Extraction		
Method	P (%)	R (%)	F (%)	P (%)	R (%)	F (%)
BLSTM	42.09	67.68	51.89	90.93	38.50	54.09
BLSTM + Word Level Attention	39.30	74.75	51.52	90.75	43.00	58.35
BLSTM + Sentence Level Attention	38.14	77.26	51.07	89.69	44.12	59.14
BLSTM + Word Level Attention + Sentence Level Attention	37.06	78.45	50.34	90.24	44.50	59.61

with many lexical features. The results of Table 5 show that the proposed model, which employs multi-level attention and dependency word embeddings, is beneficial to improve the performance of biomedical event extraction. In this work, we employ sentence vector and BLSTM based encoder to capture the accurate clues related to the events in the sentence. At each time step, the current trigger candidate is classified depending on not only the word itself and the words within a window but also other clues related to the event in the sentence, which are helpful for accurate classification. Thus, our model has higher precision with a little loss of recall than the other state-of-the-art methods.

5. Discussion

From the above experimental results, we can conclude that our BLSTM model based on multi-level attention mechanism outperforms most state-of-the-art systems and mainly includes the following important advantages:

Dependency-based Word Embeddings: The dependency-based word embeddings can obtain the dependency relations, which are important for event extraction. It is difficult to obtain the relation between words that are far apart by the traditional word embeddings. However, the dependency-based word embeddings can capture the long-distance dependency information. For example, in the sentence fragment 'The toxin prevented induction of 1L-10" (Fig. 5), the relation between "prevented" and "1L-10" is out-of-reach when the size of the window is 1 or 2. Furthermore, the "coincidental" contexts which are within the window but not directly related to the target word can be filtered out (e.g. the word "The is not used as the context for "prevented"). In addition, the contexts are typed, for example, "1L-10induction" is the object of "prevented, and "toxin is the subject. Therefore, we expect the dependency-based contexts to yield more focused embeddings, capturing more functional and less topical similarity. Table 2 shows the effectiveness of the dependency-based word embeddings.

Sentence Vector: To take advantage of both the pre-trained word embeddings and fine-tuned word embeddings, we generate sentence vectors by averaging the differences of the two word embeddings in a sentence. Sentence vectors can establish the relation between word level features and sentence level features, enrich context information. In addition, for our task, there is a strong association between events appearing in a sentence. There are multiple events in a sentence, and each event has its own triggers and arguments. The semantic information of the triggers and arguments may be helpful to identify each other. Also, there may be nested events in a sentence, which means a trigger may be the argument of the other event in the sentence (such as event E2 and E3 of Fig. 1). That means, there might be interrelation between any two words in a sentence. Therefore, the global information of the sentence is important for event extraction. Thus, we supplement the sentence vectors to capture the global sentence-level features. The experimental results reveal that the sentence vectors have the significant impact on trigger detection.

Multi-level Attention Mechanism: To filter out the irrelevant noise and find the important units of the inputs, we integrate multi-level attention for event extraction. The contribution is that it can automatically focus on the words that have decisive effect on classification and enhance the effect among relevant arguments without using extra knowledge and NLP tools.

Effectiveness of Word Level Attention: The word level attention focuses on important words within one sentence. Figure 6 shows to what extent the attentive model focuses on the contextual representations of the sentence "**Inhibition** of **angiogenesis** has been shown to be an effective strategy in the **therapy**." In the sentence, there are three triggers, which are "Inhibition", "angiogenesis" and "therapy" respectively. All the words are treated equally before integrating attention. However, after integrating the attention mechanism in our architecture, as shown in Fig. 6, most of the verbs ("has", "shown") and nouns ("Inhibition", "angiogenesis", "therapy") in the sentence are strengthened by the attention weights in the training process. The event triggers are usually verbs or gerunds, and arguments are usually nouns or other triggers (nested events). Therefore, the potential triggers and arguments might be focused on via the attention, which might be helpful for trigger detection. Also, the enhancement of the argument information might be helpful for determining the trigger types.

Effectiveness of Sentence Level Attention: The interaction among relevant arguments will be enhanced by the sentence level attention. For instance, in the "Binding" type event: *(Type: Binding, Trigger: binding, Theme: TRAF2, Theme: CD40)* in Fig. 7, the arguments *(binding, TRAF2)* and *(binding, CD40)* with the same trigger "binding" are treated as relevant arguments. After integrating the sentence level attention, the interaction among relevant arguments will play a role in the argument prediction. Therefore, if one of the above argument is predicted as a "Theme" type argument, we would have more confidence in predicting the other argument as a "Theme" type argument. That's

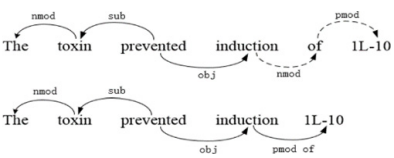

Fig. 5 Example of dependency-based context

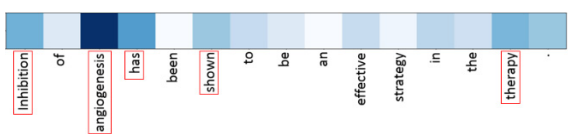

Fig. 6 Visualization of attentions over words

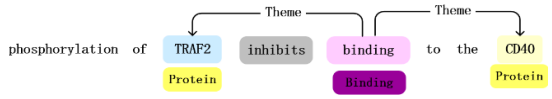

Fig. 7 An example of biomedical events

because there are usually multiple "Theme" type arguments in the "Binding" type event. The experimental results also reveal the effectiveness of the attention.

6. Conclusion

In this paper, we propose a multi-level attention mechanism based on BLSTM neural network for biomedical event extraction. The BLSTM can reduce the manual efforts and obtain both left and right context information in sentence. The dependency-based word embeddings capture more semantic and syntactic information in dependency contexts, which is beneficial for our task. Sentence vectors enrich the sentence level global features of the events within a sentence. Multi-level attention mechanism captures the most important information and enhance the interactions among relevant arguments. Our method achieves 59.61% F-score without using additional handcrafted features, and outperforms other state-of-the-art systems, which demonstrates the potential and effectiveness of the proposed framework.

Acknowledgments

The authors gratefully acknowledge the financial support provided by the National Natural Science Foundation of China under No. 61672126.

References

[1] S. Pyysalo, T. Ohta, M. Miwa, H.-C. Cho, J. Tsujii, and S. Ananiadou, Event extraction across multiple levels of biological organization, Bioinformatics, vol.28, no.18, pp.i575–i581, Sept. 2012.
[2] D. Zhou and D. Zhong, "A semi-supervised learning framework for biomedical event extraction based on hidden topics," Artificial Intelligence in Medicine, vol.64, no.1, pp.51–58, May 2015.
[3] C. Lee, W.J. Hou, and H.H. Chen, "Annotating multiple types of biomedical entities: A single word classification approach," Proc. International Joint Workshop on Natural Language Processing in Biomedicine and Its Applications (JNLPBA 04), pp.80–83, 2004.
[4] D. Zhou, D. Zhong, and Y. He, "Event trigger identification for biomedical events extraction using domain knowledge," Bioinformatics, vol.30, no.11, pp.1587–1594, Jan. 2014.
[5] J. Wang, J. Zhang, A. Yuan, Y. An, H. Lin, Z. Yang, Y. Zhang, and Y. Sun, "Biomedical event trigger detection by dependency-based word embedding," IEEE Int. Conf. Bioinformatics and Biomedicine, pp.429–432, 2015.
[6] Y. Nie, W. Rong, Y. Zhang, Y. Ouyang, and Z. Xiong, "Embedding assisted prediction architecture for event trigger identification," J. Bioinformatics & Computational Biology, vol.13, no.3, pp.575–577, 2015.
[7] A. Wang, J. Wang, H. Lin, J. Zhang, Z. Yang, and K. Xu, "A multiple distributed representation method based on neural network for biomedical event extraction," Bmc Medical Informatics & Decision Making, 17(Suppl 3):171, Dec. 2017.
[8] O. Levy, and Y. Goldberg, "Dependency-based word embedding," Meeting of the Association for Computational Linguistics, Baltimore, Maryland, USA, June 2014.
[9] M. Miwa, S. Pyysalo, T. Hara, and J. Tsujii, "Evaluating dependency representation for event extraction," International conference on computational linguistics, pp.779–787, Aug. 2010.
[10] K. Sagae and J. Tsujii, "Dependency Parsing and Domain Adaptation with Data-Driven LR Models and Parser Ensembles[M]," Springer Netherlands, 2007.
[11] L. Li, L. Jin, Y. Jiang, and D. Huang, "Recognizing biomedical named entities based on the sentence vector/twin word embeddings conditioned bidirectional LSTM," China National Conference on Chinese Computational Linguistics, ShanDong, Yantai, China, 2016.
[12] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Computation, vol.9, no.8, pp.1735–1780, Nov. 1997.
[13] S. Zhang, D. Zheng, X. Hu, and M. Yang, "Bidirectional long short-term memory networks for relation classification," Proc. Pacific Asia Conference on Language, Information and Computation, ShangHai, China, 2015.
[14] V. Mnih, N. Heess, A. Graves, and K. Kavukcuoglu, "Recurrent models of visual attention," Advances in Neural Information Processing Systems, Montreal, Quebec, Canada, 2014.
[15] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," Computer Science, arXiv preprint arXiv:1409-0473, 2014.
[16] A.M. Rush, S. Chopra, and J. Weston, "A neural attention model for abstractive sentence summarization," Computer Science, arXiv:1509.00685, 2015.
[17] J. Björne, J. Heimonen, F. Ginter, A. Airola, T. Pahikkala, and T. Salakoski, "Extracting complex biological events with rich graph-based feature sets," Computational Intelligence, vol.27, no.4, pp.541–557, 2009.
[18] J. Björne, "Biomedical Event Extraction with Machine Learning," TUCS Dissertations, vol.178, pp.1–121, 2014.
[19] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," J. Mach. Learn. Res., vol.15, no.1, pp.1929–1958, 2014.

Xinyu He received her BSc degree from Shenyang Normal University in 2006, and her MSc degree from Dalian University of Technology in 2012. She is an Ph.D. candidate in the School of Computer Software and Theory at Dalian University of technology. Her research interests include text mining for biomedical literatures and information extraction.

Lishuang Li received her BSc degree, MSc degree and Ph.D. degree from Dalian University of Technology in 1989, 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 70 research papers. Her research interests include natural language processing, deep learning, text mining for biomedical literatures and information extraction. Her research projects are funded by the NSFC.

Xingchen Song has been majoring in Computer Science and Technology at Dalian University of Technology for his Bachelor of Engineering degree since 2015. His research interests include machine learning and Natural Language Processing.

Degen Huang was born in 1965. He is a professor in the Dalian University of Technology. His main research interests include natural language processing, machine learning and machine translation. He is now working at the School of Computer Science and Technology, Dalian University of Technology. He is now a senior member of CCF, and an associate editor of Int. J. Advanced Intelligence.

Fuji Ren Ph.D. Professor. His main research interests include Natural Language Processing, Knowledge Engineering, Sentience Computer, Machine Translation, Machine-Aided English Writing, Automatic Abstracting, Dialogue machine translation, Information Retrieval.