Computers, Materials & Continua CMC, vol.63, no.2, pp.923-941, 2020

Biomedical Event Extraction Using a New Error Detection Learning Approach Based on Neural Network

Xiaolei Ma[1, 2], Yang Lu[1, 2], Yinan Lu[1, *], Zhili Pei[2] and Jichao Liu[3]

Abstract: Supervised machine learning approaches are effective in text mining, but their success relies heavily on manually annotated corpora. However, there are limited numbers of annotated biomedical event corpora, and the available datasets contain insufficient examples for training classifiers; the common cure is to seek large amounts of training samples from unlabeled data, but such data sets often contain many mislabeled samples, which will degrade the performance of classifiers. Therefore, this study proposes a novel error data detection approach suitable for reducing noise in unlabeled biomedical event data. First, we construct the mislabeled dataset through error data analysis with the development dataset. The sample pairs' vector representations are then obtained by the means of sequence patterns and the joint model of convolutional neural network and long short-term memory recurrent neural network. Following this, the sample identification strategy is proposed, using error detection based on pair representation for unlabeled data. With the latter, the selected samples are added to enrich the training dataset and improve the classification performance. In the BioNLP Shared Task GENIA, the experiments results indicate that the proposed approach is competent in extract the biomedical event from biomedical literature. Our approach can effectively filter some noisy examples and build a satisfactory prediction model.

Keywords: Biomedical event extraction, pair representation, error data detection, sample identification.

1 Introduction

PubMed is one of the largest and most widely-used electronic medical literature resources. The medical literature in PubMed grows at approximately two pages per second, and there are currently at least 26 million articles in PubMed [Lu (2011)]. In the face of this rapid growth of unstructured literature, the number of topics that are of interest to researchers will surpass their ability to personally read and vet each potentially relevant article.

[1] College of the Computer Science and Technology, Jilin University, Changchun, 130012, China.

[2] College of the Computer Science and Technology, Inner Mongolia University for Nationalities, Tongliao, 028000, China.

[3] School of Science and Technology, Yanching Institute of Technology, Langfang, 065202, China.

* Corresponding Author: Yinan Lu. Email: luyn_jlu@sina.com.

Received: 21 June 2019; Accepted: 04 November 2019.

Therefore, the ability to automatically extract biomedical information would be helpful to researchers. Although many tools or methods for named entity recognition and relation extraction exist in the biomedical field, this simple approach is unable to meet researchers' needs. Instead, extracting an understanding of biomedical events and their descriptions from biomedical literature is necessary. A biomedical event is a process of molecular interactions; finding them involves extracting the semantic and role information of biological events. Thus, accurately and effectively extracting complex biomedical events is a great challenge. Consequently, the possibility of automatically extracting biomedical events from large volumes of biomedical text has attracted increasing attention.

Early detection of biomedical events is a simple process that extracts pairwise relations between entities, such as interactions between drugs (DDI) [Yamazaki (2018)], interactions between proteins (PPI) [Antonov, Dietmann, Rodchenkov et al. (2009)], and relationships between genes and disease [Lee, Ahmed, Loriot et al. (2018)]. However, these simple relationships are insufficient to represent the more complex relations often encountered in real-world situations. Therefore, a series of challenges entitled BioNLP Shared Task (BioNLP-ST) [Kim, Ohta, Pyysalo et al. (2009)] was formulated starting in 2009 by the BioNLP special interest group. The goal of these tasks is to extract rich, complex, structured biological process relationships from biomedical texts. The most important of these tasks is GE, which represents dynamic biological processes that involve a change in location or interactions between entities, such as genes, cells, and some chemicals.

In general, a biomedical event extends binary relations by adding to their types and nesting. A binary relation consists of a trigger and one or more arguments. Most triggers are verbs (although a few are nouns) that cause interesting events. The arguments are the entities that participate in such events. A biomedical event extraction system must be able to identify triggers, their corresponding arguments, and the type of event to which they belong. For example, one biomedical event type in BioNLP-ST GENIA Event Extraction 2011 (GE'11) is divided into nine categories and was further extended to fourteen categories in BioNLP-ST GENIA Event Extraction 2013 (GE'13), such as gene_expression, transcription, localization, protein_catabolism and phosphorylation. These five events are called simple events (SVT) because each event has only one theme as its argument. In contrast, the three events regulation, positive_regulation, and negative_regulation may have complex structures that may include both a theme and an optional cause as their arguments; these typify a regulation event (REG). A binding is called a BIND event and may have two arguments. The complexity of such events can be demonstrated as an example: given the sentence "Sp1 and Sp3 regulate basal transcription of human APOBEC3G gene", a biomedical event extraction system should extract the events shown in Fig. 1 and listed below.

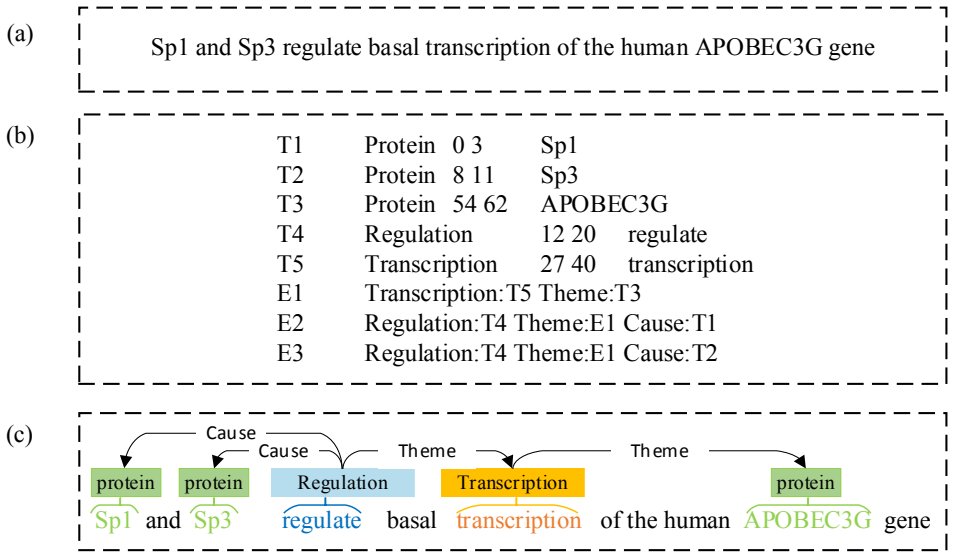


Figure 1: Examples of biomedical events. (**a**) an example of sentences. (**b**) annotated event examples of the given sentence. (**c**) event examples visualization

Supervised machine learning (SML) has been widely used in biomedical event extraction. The Turku system [Björne, Ginter and Salakoski (2012)] and the EVEX system [Kai, Landeghem, Salakoski et al. (2013)] use machine learning (ML) to extract biomedical events. Their approach relies on the pipeline model, the process of which can be divided into three phases: trigger identification, argument assignment and event element detection. Their pipeline models have achieved excellent results, but are subject to errors because each phase is conducted based on the previous step; thus, when a prior step obtains an incorrect result, the subsequent steps will also be incorrect—a situation known as "cascading errors." The problem of cascading errors was overcome by the joint model [Björne, Heimonen, Ginter et al. (2009); Miwa, Saetre, Kim et al. (2010)]. Although joint models perform well, their computational cost is high because they regard all word combinations as possible events. The pairwise model [Özgür and Radev (2009); Xiao, Bordes and Grandvalet (2013)] is a combination of the pipeline and joint models that solves some of their shortcomings. The authors of Hou et al. [Hou and Ceesay (2015); Kolya, Ekbal and Bandyopadhyay (2012)] manually constructed suitable patterns for extracting biomedical events; the resulting models are called rule-based models. All the abovementioned systems are supervised learning approaches.

However, one critical issue is not addressed well by the systems mentioned above. Annotated corpora are limited and imbalanced; they are insufficient to fully train a model, which may limit system performance. Manually constructing annotated corpora is a time- and labor-intensive task. To solve this problem, one feasible solution is to use large-scale unlabeled corpora because such unlabeled data are always easier to obtain. Wang et al. [Wang, Xu, Lin et al. (2013)] designed rich features that improved the accuracy of the trigger identification of biomedical event extraction using a semisupervised method. Zhou et al. [Zhou and Zhong (2015)] proposed a method that could automatically assign event

type by calculating the distance between sentences from unlabeled corpus and the sentences in the annotated corpus. Although the semisupervised method has its advantages, with the increase of training times, noise in the data is increasing and classification performance is decreasing, which requires an effective method to address this problem. Deep neural network models are often used to generate high-level semantic vectors and classification. Xiong et al. [Xiong, Shen, Wang et al. (2018)] used a new model which combines the CBOW model and CNN to generate sentence and paragraph vectors for the task of natural language processing.

In this paper, we propose an error detection pair representation-based (EDPR) method to solve the problems mentioned above in biomedical event extraction. Our method is an iterative learning process using self-training (ST). First, we build a mislabeled dataset and generate sequential patterns from the mislabeled dataset to determine the patterns of those mislabeled samples. We then present the pattern-based vector representation of pairs, which is obtained by the means of convolutional neural network (CNN), long short-term memory recurrent neural network (C-LSTM), and sequence patterns. Following this, we design a sample identification strategy to remove those noisy samples which have a negative effect on the subsequent learning process. With the latter, the selected samples are added to enrich the training dataset and improve the classification performance.

2 Materials and methods

2.1 Text processing

The text is preprocessed using natural language tools such as a parser, etc. For example, here, the labeled training data are analyzed by tokenization, sentence splitting and dependence parsing. We used the set of features proposed by Xiao et al. [Xiao, Bordes and Grandvalet (2013)] for classification. The features are as follows:

Candidate entity features include base features, such as stem, part-of-speech (POS) and *n*-gram features (n = 2,3,4). Also, the base features of neighborhood around the candidate entity.

Argument features include base features, the context around the argument and knowledge base features when the argument is a protein.

Pairwise features include base features between candidate and argument, shortest dependency path features. Such as the E-walk (dep-tag, token, dep-tag) and V-walk (token, dep-tag, token) features between candidate and argument over the shortest path, where tokens are stem and POS tags, and dep-tags are the dependency labels. Besides, token sequence feature over the shortest path.

The output of the text preprocessing step is used as the input for the subsequent step.

2.2 Definitions in learning process

The initial labeled dataset includes k event types and n samples is denoted as $D_L = \{x_i\}_{i=1}^{n}$, let the label of x_i be $y_i = j, j \in \{0,1,\dots,k\}$, where x_i belongs to the event type j. Among the initial labeled samples, those in the training dataset are denoted as D_{train}, $D_{train} \subset D_L$, while those in the development dataset are denoted as D_{dev}, $D_{dev} \subset D_L$. Each sample x_i is represented as a pair (trigger, argument), where the trigger and argument exist in the same sentence. A base linear classifier F is trained on D_L, next the unlabeled dataset D_U is

labeled using F. Then, the self-training method is applied to selects appropriate samples to add them into the training set. Our method is an iterative learning process until the maximum iteration number is satisfied.

2.3 Sample identification based on error detection

During the self-training process, with the unlabeled data which labeled incorrectly by the initial classifier incorporating into the training dataset, the classification performance is dramatically degraded. To solve this problem, we propose the EDPR method to identify the mislabeled samples. The purpose of EDPR is to ensure the credibility and integrity of predicted events. The training process of the proposed method are depicted in Fig. 2. We first build a mislabeled dataset and generate sequential patterns from the mislabeled dataset. Further, the sample pairs vector representations are obtained by the means of C-LSTM model and sequence patterns. Finally, a sample identification strategy based on EDPR analysis method is presented.

2.3.1 The construction of mislabeled dataset

In this study, we construct a mislabeled dataset. First, a linear classifier F_0 is trained on D_{train}, which is used to predict a class label for each item in D_{dev}. Then, the pseudo labeled development dataset is obtained, which is denoted as D_{pse}. Second, for each sample x_{pse} in D_{pse}, if the corresponding label y_{pse} is different as it appears in raw D_{dev}, the sample x_{pse} is considered as mislabeled. Then, the mislabeled dataset is obtained denoted as D_{mis}.

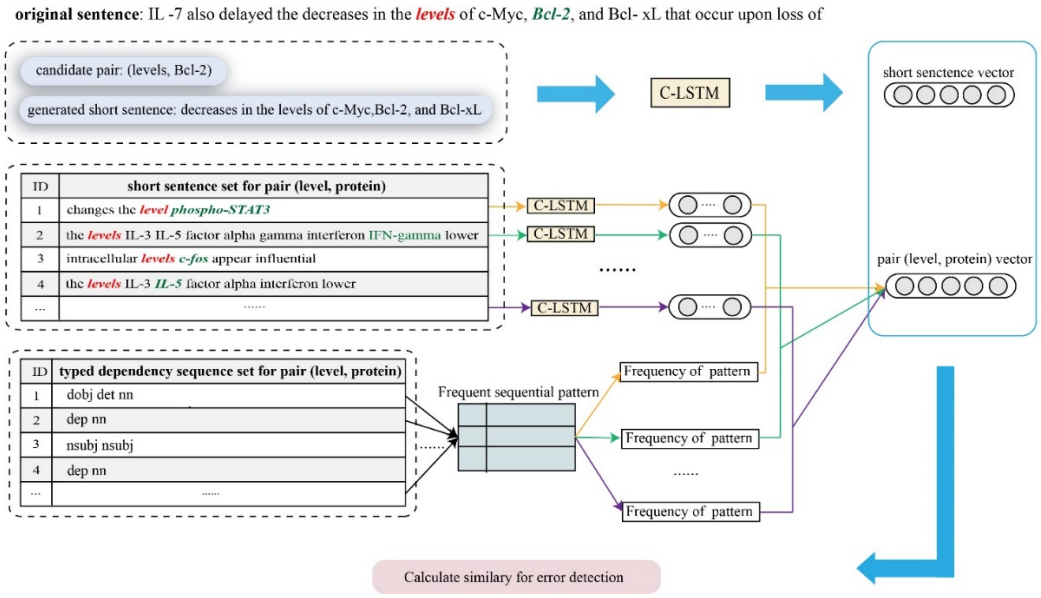


Figure 2: An overview of training process in EDPR

2.3.2 Pattern generation

In the pattern generation phase, the mislabeled dataset is used for pattern extraction. Initially, the typed dependency sequence generated from the shortest dependency path between pairs is denoted as $D_{mis}(s) = \{s_n | s_1, s_2, \cdots, s_n\}$. For example, the sequence $s_1 = < amod, prep_to, prep_in, nn >$is a typed dependency sequence. Following this, the *PrefixSpan* algorithm [Pei, Han, Mortazavi-Asl et al. (2002)] is applied to $D_{mis}(s)$. And the frequent sequential pattern set is obtained. Sequential pattern mining aims to find frequent sub-sequences that satisfy the minimum support. Tab. 1 shows part of frequent sequential patterns, and the minimum support is set 2.

Table 1: Part of frequent sequential patterns

ID	Sequence database	Frequent sequential pattern
s_1	<prep_through, xsubj, nn>	
s_2	<amod, prep_to, prep_in, nn>	<prep_in, nn>
s_3	< dobj nsubj prep_in nn>	<amod, nn>
s_4	<amod, nn>	

2.3.3 Pairs representation

A C-LSTM model-based vector representation of pairs is proposed in this study. The shortest dependency path parser is one of famous for syntactic analysis in biomedical event extract. Pairs vector representations make use of the shortest dependency path between pairs, which can capture rich semantic information. We expand the shortest dependency path by adding the subtrees to obtain more information, which was proposed by Yang et al. [Yang, Wei, Li et al. (2015)]. An example of expending shortest dependency path for pair (expression, IRF4) as follow:

Raw Sentence: *Absence of IRF-4 expression in leukemia cells is not due to promoter alterations.*

The word sequence of shortest dependency path: *expression IRF-4.*

After expended word sequence: *absence of IRF-4 expression in cells.*

The expended word sequence is denoted as $S =< w_1, w_2, \cdots, w_n >$for a pair. Next, a sequence of vectors $p = [x_1, x_2, \cdots, x_L]$ for S is obtained by pre-training word vectors [Pyysalo, Ginter, Moen et al. (2013)], which gained from published materials PubMed-and-PMC-w2v word embedding (http://evexdb.org/pmresources/vec-space-models/). And the p as raw input for the input layer of C-LSTM. The pair vector representation $\vec{v}_p$ is then encoded by applying C-LSTM model.

Neural network models can learn powerful features and have been achieved excellent results in sentence and text modeling. The C-LSTM model which combines convolution neural network with long short-term memory network is adopted in this paper. We followed the work of Zhou et al. [Zhou, Sun, Liu et al. (2015)] to build our C-LSTM model. They proposed the C-LSTM model can learn phrase-level features through convolutional layer and are fed into the LSTM to obtain the sentence representation.

1) Convolution Neural Network

The C-LSTM model applies CNN to learn higher-level window features. Given a candidate pair (t_i, a_j), the corresponding expended word sequence is $S = \{w_1, w_2, \cdots, w_n\}$. In the input layer, the sequence of vectors p as raw input for CNN. Let $x_i \in R^d$ correspond to the *d*-dimensional word vectors for w_i in an input S (*d* is equal to 200). Then we get the word sequence matrix $M = [x_1, x_2, \cdots, x_L] \in R^{L \times d}$, where L is the length of the sequence. In addition, if the length of input sequence has less than *L*, we pad zero vectors at the end (we set *L*=30). In the convolutional layer, a convolution operation is applied to produce a new feature through a filter. Let k be the length of the filter $\mathrm{W} \in R^{k \times d}$. A window vector with k consecutive word vectors can be constructed to a matrix $M_{i,i+k-1} = [x_i, x_{i+1}, \cdots, x_{i+k-1}] \epsilon R^{k \times d}$. Filter W convolves with matrix $M_{i,i+k-1}$ generate a feature c_i as follow:

$$c_i = f(W \cdot M_{i,i+k-1} + b) \tag{1}$$

where $b \in \mathrm{R}$ is a bias term and f is a non-linear function, we choose the Rectified Linear Unit (Relu) as the non-linear function. Then, a feature map c is generated through a filter convolves with the k-window vectors at each position.

$$\mathrm{c} = [c_1, c_2, \cdots, c_{L-k+1}] \tag{2}$$

After the convolutional layer, max-pooling operation is not applied to feature maps to extract the most important feature, instead, LSTM is stacked on the top of the CNN. LSTM is able to capture continuous features. Sequences of such higher-level representations are then fed into the LSTM to learn long-term dependencies.

2) Long Short-Term Memory Networks

LSTM to learn sequential correlations from higher-order sequential features. Here, the LSTM model describe the implementation used by the standard architecture [Hochreiter and Schmidhuber (1997)]. x_t is the current input. At each time step, an old hidden state h_{t-1}, an input gate i_t, a forget gate f_t , an actual input and an output o_t, these gates control the information flow for current memory cell c_t and hidden state h_t. The LSTM transition equations are the following:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{3}$$

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{4}$$

$$u_t = \tanh(W_q \cdot [h_{t-1}, x_t] + b_q) \tag{5}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{6}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot u_t \tag{7}$$

$$h_t = o_t \odot \tanh(c_t) \tag{8}$$

where $\odot$ denotes element-wise multiplication, σ is the logistic sigmoid function that make the gating values in [0,1]. LSTM is useful for learn long-term dependencies in sequences. In this paper, we regard the output of the hidden state at the last time step of LSTM to represent the vector of a pair.

2.3.4 Error detection approach

The mislabeled dataset D_{mis} is divided into *k*-class subsets based on their event types at first, where $D_{mis} = \{DS_{mis}^{m}\}_{m=1}^{k}$. Given a mislabeled pair $P_{mis\_ij}(t_i, a_j)$, $P_{mis\_ij}(t_i, a_j) \in DS_{mis}^{m}$, $m \in k$, if the argument is a protein, we replace the protein name as keyword "protein". For example, the pair (express, IRF4) is represented as (express, protein). There exist a variety of extended word sequence in different sentences for pair $P_{mis\_ij}(t_i, a_j)$. Let the extended word sequence set for the pair $P_{mis\_ij}(t_i, a_j)$ is $S_{P_{mis\_ij}} = \{s_{ij}\}_{ij=1}^{n}$. Then each sequence s_{ij} of the pair $P_{mis\_ij}(t_i, a_j)$ sequence set is feed into the C-LSTM model to obtain sequence vector $\vec{v}_{s_{ij}}$. For each mislabeled pair $P_{mis\_ij}(t_i, a_j)$, the vector representation employs weighted mean method is computed as follows:

$$\vec{v}_{p_{mis\_ij}} = \frac{\sum_{s_{ij} \in S_{P_{mis\_ij}}} \vec{v}_{s_{ij}} \cdot f_{s_{ij}}}{\sum_{s_{ij} \in S_{P_{mis\_ij}}} f_{s_{ij}}} \tag{9}$$

$\vec{v}_{p_{mis\_ij}}$ represents the vector of the mislabeled pair $P_{mis}(t_i, a_j)$. Let the frequent patterns set is FS, $f_{s_{ij}}$ is the number of sequences that s_{ij} contains in FS. For example, let sequences FS_1 = {amod, prep_to, nn}, FS_2={amod, nn} and FS_3={prep_to, nn} are the three frequent sequences in FS. For the sequence s_{ij} = {amod, prep_to, prep_in, nn}, it contains the three frequent sequences, so $f_{s_{ij}}$=3. For each predicted pair $P_{pre\_gh}(t_g, a_h)$, use C-LSTM model to obtain sequence vector $\vec{v}_{s_{gh}}$, where s_{gh} represents the sequence between the trigger t_g and argument a_h. If the a_h is a protein, we replace the a_h as keyword "protein". Then calculated similarity of mislabeled and predicted pair corresponding to the same class based on pair vector. The formula is given as follows.

$$Sim(P_{pre\_gh}, P_{mis\_ij}) \begin{cases} \cos(\vec{v}_{s_{gh}}, \vec{v}_{p_{mis\_ij}}) & if\ g = i \text{ and } h = j \\ 0 & otherwise \end{cases} \tag{10}$$

Given a threshold α, if the $Sim(P_{pre\_gh}, P_{mis\_ij}) > \alpha$, this predicted pair is considered as mislabeled sample. Algorithm 1 summaries the learning process of the proposed method.

Algorithm 1

1: **Initialize**:
2: Given an initial labeled dataset D_L, $D_L = D_{train} \cup D_{dev}$, and unlabeled dataset D_U.
3: Maximum iteration number *t*
4: Train an initial classifier F_0 using D_{train}
5: D_{mis} = Data_process ()
6: **while (not reach maximum iteration *t*) do**
7: **if** $t = 0$ **then**
8: $F = F_0$
9: **else**

Train classifier F using D_L
end if
Select a batch of candidate samples D_{batch} from D_U.
Obtain the prediction set D_{cand} by current classifier F using D_{batch}.
$D_{certain}$=Error_Detection (D_{cand})
Update $D_L = D_L + D_{certain}$, $D_U = D_U - D_{batch}$.
end while
procedure Data_process ()
Initialize $D_{mis} = \emptyset$.
Use F_0 to predict class label of D_{dev}, and obtain the pseudo labeled dataset D_{pse}.
Given $(x_{dev}, y_{dev}) \in D_{dev}$
for each $(x_{pse}, y_{pse}) \in D_{pse}$ **do**
if (x_{pse}, y_{pse}) in D_{dev} **then**
if $y_{pse} != y_{dev}$ **then**
$D_{mis} = D_{mis} \cup (x_{pse}, y_{pse})$
end if
end if
end for
Divide D_{mis} into k subsets based on event types, $D_{mis} = \{DS_{mis}^m\}_{m=1}^k$.
return D_{mis}.
end procedure
procedure Error _Detection (D_{cand})
Obtain pair vector $\vec{v}_{p_{mis\_ij}}$ of $P_{mis\_ij}(t_i, a_j)$ using Eq. (9),$P_{mis\_ij} \in DS_{mis}^m$, $DS_{mis}^m \subset D_{mis}$
Select samples set $D_{certain}$ based on pair vector using Eq. (10), $D_{certain} \subset D_{cand}$
return $D_{certain}$.
end procedure

3 Results

3.1 Experimental setup

In this section, we evaluate our proposed EDPR approach on the GE'11 and GE'13 corpora. We use a linear support vector machine (SVM) with a "one-vs-the-rest" multiclass strategy

as the base classifier and adopt the Charniak-Johnson parser with the biomedical parsing model of McClosky et al. [Mcclosky, Surdeanu and Manning (2011)] to create dependency features. Nine types of events are defined in GE'11 but fourteen types of events are defined in GE'13. Because very few samples of the newly defined event types in GE'13 are available, this study uses only the nine event types defined in GE'11 for this evaluation. In our experiment the unlabeled corpora including proteins annotations (bioconcepts2pubtator_offsets.gz) are downloaded from PubTator (ftp://ftp.ncbi.nlm.nih.gov/pub/lu/PubTator/) [Wei, Kao and Lu (2013)]. We limited the full learning process to 5 iterations, and 300 articles per iteration, the threshold of a sample identification strategy is 0.6. During each iteration, the samples meet the criteria of EDPR will be added to the training dataset automatically. An official online assessment tool is applied to optimize all the parameters on the development set, and all our experimental results are reported as *approximate span, recursive*.

3.2 Experiment results

To verify the effectiveness of our proposed EDPR method, we conduct the experiments with GENIA dataset. Tab. 2 provides details of the GE'11 test results, the F-score of SVT event class reaches 73.82, while that of Bind event class is 52.10 and that of REG event class is 45.19, and the total F-score of 55.74 achieves good results. Tab. 3 gives the results of the experiment using the proposed method on GE'13 test set.

Table 2: Results of our method on test set of GE'11

Event class	Event type	Rec. (%)	Prec. (%)	F (%)
SVT	*Gene_expression*	75.15	83.39	79.06
	Transcription	53.45	70.45	60.78
	Protein_catabolism	80.00	80.00	80.00
	Phosphorylation	69.19	86.49	76.53
	Localization	31.41	86.96	46.15
	TOTAL	66.75	82.56	73.82
BIND	Binding	51.73	52.48	52.10
REG	*Regulation*	41.82	47.21	44.35
	Positive_regulation	40.40	52.33	45.60
	Negative_regulation	48.51	41.65	44.82
	TOTAL	42.56	48.16	45.19
ALL TOTAL		52.08	59.96	55.74

Performance is shown in Recall (REC.), Precision (Prec.) and F-score (F).

Table 3: Results of our method on test set of GE'13

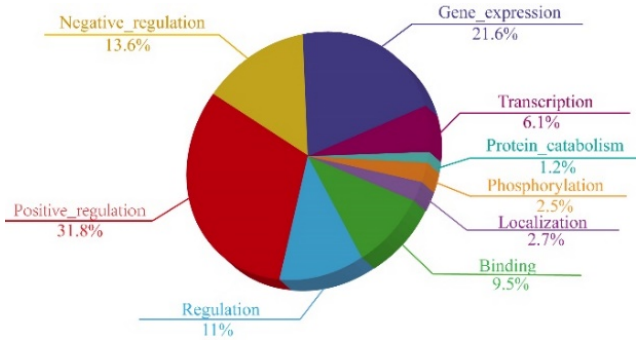
Event class	Event type	Rec. (%)	Prec. (%)	F (%)
SVT	*Gene_expression*	81.42	85.71	83.51
	Transcription	49.50	71.43	58.48
	Protein_catabolism	57.14	57.14	57.14
	Phosphorylation	78.75	78.75	78.75
	Localization	27.27	87.10	41.54
	TOTAL	72.00	82.85	77.04
BIND	Binding	40.24	44.82	42.41
REG	*Regulation*	23.61	43.87	30.70
	Positive_regulation	37.52	61.54	46.62
	Negative_regulation	42.40	47.55	44.83
	TOTAL	36.78	54.46	43.91
ALL TOTAL		47.83	63.20	54.45

Performance is shown in Recall (REC.), Precision (Prec.) and F-score (F).

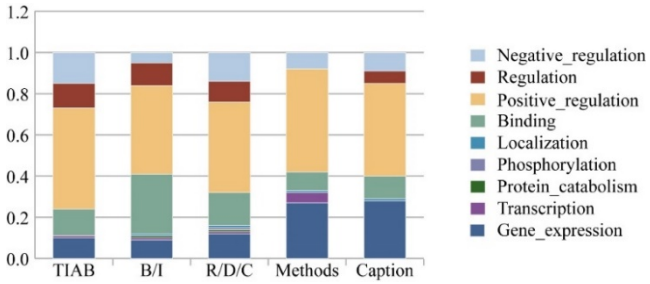
3.3 Experiment analysis and performance evaluation

This section gives a more detailed analysis of performances of the proposed method. To better build a mislabeled dataset, we analyze the proportion of each event in the training set, and the different distribution of error samples of each event in the five section groups of GE'13 development (GE'13 development provides online error analysis). Fig. 3(a) shows the proportion of each event in the training set, where positive_regulation accounts for 31.8% of the total, followed by Gene_expression, Negative_regulation, Reuglation and Binding, which is 21.6%, 13.6%, 11% and 9.5% respectively. And we can see from Fig. 3(b) that the error rate of Positive_regulation, Gene_expression, Binding, Neagtive_regulation and Regulation are the largest in the five section groups of GE'13, other events with less error rate in Fig. 3(b) have less proportion in Fig. 3(a). Therefore, when we build the mislabeled dataset, we only consider the aforementioned events, which have more mislabeled samples, because too few samples of event are not conducive to the next stage of pair representation, thus affecting the predictive performance of the classifier in this event.

Sample identification process for some examples are summarized in Tab. 4. For example, the first row shows that a candidate pair (positive, CD40) in Sentence ID 8 of PMID-10096561 is predicted to Gene_expression event type. We will find generated vector for pair (positive, protein) by the proposed method in the mislabeled database, and then compare the similarity with the corresponding candidated sample (positive, CD40). The similarity score of the pair (positive, protein) and candidatedsample (positive, CD40) is 0.89, which is higher than the threshold 0.6. We consider that the candidated sample is noise and it will be discarded and not added to the training set. The event type of the pair (positive, CD40) in the development set is None in fact. Therefore, this EDPR method is effective.



(a)



(b)

Figure 3: (**a**) Event proportion in training set. (**b**) Error event distribution in different sections of GE'13 development set

Table 4: Examples of error event

Document ID	Sentence ID	Sample pair	Short Sentence	Similarity Score	Predicted Type	True Type
PMID-10096561	8	positive, CD40	some CD40 positive immunogenic human MMs	0.89	Gene_expression	None
PMID-9796702	11	in response to, activated	PKB is activated in response to triggering required sufficient	0.85	Positive_regulation	None
PMID-9796963	5	altered, expression	characterize altered expression TCRzeta activation	0.79	Regulation	Negative_ regulation
PMID-10415075	0	expression, p65-RelA	associated decreased p65-RelA protein expression	0.8	Gene_expression	None
PMID-10096561	4	triggering, CD40	stimulation beta CD40 triggering	0.92	Positive_regulation	None

To illustrate the effectiveness of EDPR, we compare EDPR, the original model and original model with ST (Original-ST) on the GE'11 development set; the paper shows only the results from 300, 600, 900, 1200 and 1500 articles. As show in Fig. 4, during the iteration process, the F-score of the green and blue curves show a trend of first rising and then falling, but the F-score of green curve is always higher than that of blue curve. The

phenomenon shows that the approach of EDPR is more effective under the same semi-supervised condition. Tab. 5 provides the number of samples added by EDPR and Original-ST model during these five iterations. Obviously, EDPR adds fewer samples than Original-ST model in each iteration process, this also shows that the proposed method can effectively remove noise and achieve better performance of the classifier.

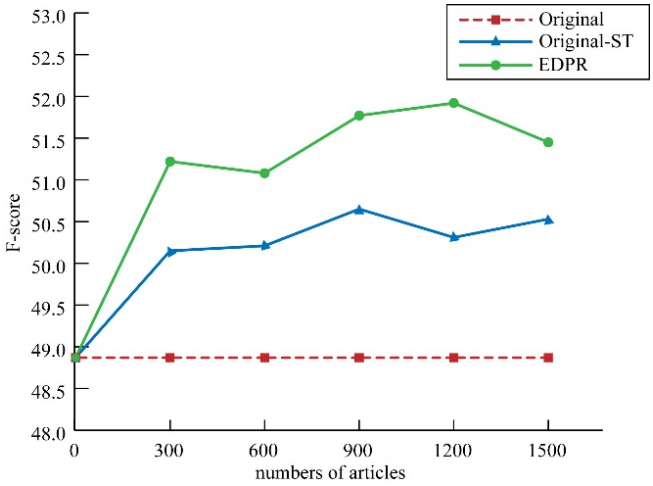


Figure 4: Event proportion in training set

Table 5: Statistics on the added samples

Numbers of articles method	0	300	600	900	1200	1500
Original	11419	-	-	-	-	-
Original-ST	11419	13456	15314	17206	19104	21466
EDPR	11419	12705	13882	15069	16266	17366

To further demonstrate the effectiveness of EDPR, we present a detailed analysis comparing EDPR, original model with and without ST on GE'11 (GE'13) development set when they are optimal in the iteration process. As shown in Fig. 5, we find that the F-score of each event of EDPR and original -ST model is higher than that of original model without ST. However, those less proportionate event in the training dataset do not improve significantly. Two reasons to cause the problem have been studied, 1) the classifier itself has a high accuracy in predicting those events; 2) we do not build these events into the mislabeled dataset. In addition, although the Binding event do not achieve the desired results, which may be caused by its particularity. A possible explanation for this is that Binding event may have one or two arguments which is protein (see Section 1). However, the proposed method is pairwise-based, that is to say, all our work is done in the case of extracting pair (trigger, argument). Although Binding event accounts for a larger proportion in training set (see Fig. 3(a)), we do not take into account the case of Binding event including two arguments, which would result in too few samples of Binding event in

the mislabeled dataset. The data in Tab. 6 can illustrate this problem, take GE'13 development set for example, the number of Binding events including one argument (trigger, argument) is 187, while the number of Binding events including two arguments (trigger, argument, argument) is 215. Therefore, the number of error samples we generated is insufficient in mislabeled dataset, which leads to the less improvement of Binding event. Nevertheless, the effect of REG event (positive_regulation, regulation and negative_regulation) which is complex and the most challenge event type is very obvious.

Table 6: Statistics of Binding event with different argument in training and development set

Item	Training		Devel	
	GE'11	GE'13	GE'11	GE'13
Binding (Theme(P))	708	91	190	187
Binding (Theme(P)+)	280	104	185	215
Total	988	195	375	402

Theme(P) denotes one parameter, Theme(P+) denotes two parameters.

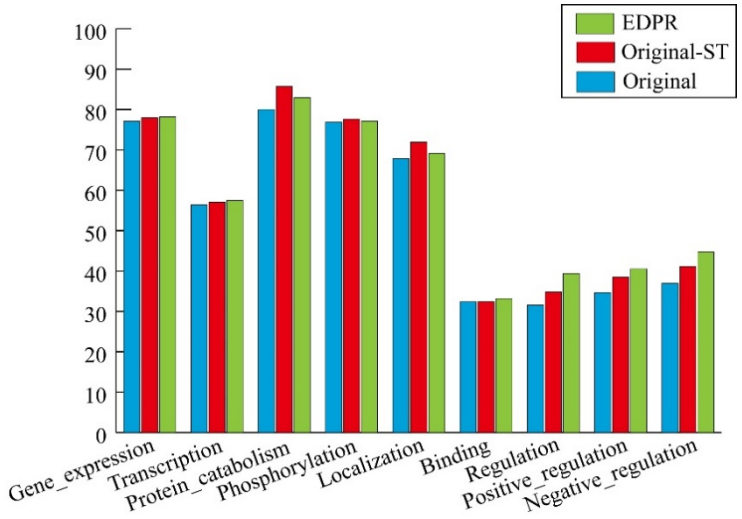


Figure 5: Comparison the optimal F-score of EDPR (error detection pair representation-based) and Original model (original labeled dataset) for each event on the GE'11 development set

3.4 Performance comparision with different methods on gene corpus

Tab. 7 shows the results of comparisons with other systems reported in Lu et al. [Lu, Ma, Lu et al. (2016); Kim, Wang, Takagi et al. (2011); Munkhdalai, Namsrai and Ryu (2015)]. As shown, our proposed method achieves a better performance than do the other systems with different types of algorithms. For example, UMass is rule-based, UTurku, MSR-NLP and study [Lu, Ma, Lu et al. (2016)] involve SML methods, and Research [Mehryary, Kaewphan, Kai et al. (2016)] is a combination of supervised and unsupervised approach. The overall F-score of EDPR for all event types is 55.74, which is slightly higher than

UMass (approximately 0.54 points) a rule-based system. Moreover, it is higher than the other four machine learning systems for all event types (Research [Mehryary, Kaewphan, Kai et al. (2016)] is 54.71, Study [Lu, Ma, Lu et al. (2016)] is 53.81, UTurku is 53.30 and MSR-NLP is 51.50). In addition, almost every value obtained from our approach performs well on the various event classes. These results demonstrate that our approach EDPR is effective on the GE'11 test set.

Table 7: Performance comparison with other systems on GE'11 test set

System	SVT	BIND	REG	All
	Rec./Prec./F-score	Rec./Prec./F-score	Rec./Prec./F-score	Rec./Prec./F-score
Ours	66.75/**82.56**/73.82	**51.73**/52.48/**52.10**	**42.56**/48.16/**45.19**	**52.08**/59.96/**55.74**
UMass	67.01/81.40/73.50	42.97/56.42/48.79	37.52/**52.67**/43.82	48.49/**64.08**/55.20
Research [Mehryary, Kaewphan, Kai et al. (2016)]	-	-	-	48.78/62.27/54.71
Study [Lu, Ma, Lu et al. (2016)]	68.60/80.34/**74.01**	47.66/**56.52**/51.71	38.97/43.88/41.28	50.35/57.79/53.81
UTurku	68.22/76.47/72.11	42.97/43.60/43.28	38.72/47.64/42.72	49.56/57.65/53.30
MSR-NLP	**68.99**/74.30/71.54	42.36/40.47/41.39	36.64/44.08/40.02	48.64/54.71/51.50

Performance is shown in Recall (REC.), Precision (Prec.) and F-score (F).

The GE'13 test set contains only full text and does not include abstracts, which is different from the GE'11 test data. Therefore, it is most "realistic" and the most challenging dataset. To evaluate the performance of the proposed method, we compared our method's results with those of EVEX, TEES 2.1, BIOSEM, NCBI and Study [Lu, Ma, Lu et al. (2016)] on the GE'13. The other results were reported in Kim et al. [Kim, Wang and Yasunori (2013)]. EVEX and TEES 2.1 are typical SVM-based pipleline model, BIOSEM uses a ruled-based method, Study [Lu, Ma, Lu et al. (2016)] is also pairwise model like us and NCBI adopts the method of joint model. From Tab. 8, we can see that the Precision and F-score of ALL of proposed approach are the highest in all systems, but the performance on BIND is far worse than that on BIOSEM which is the best on BIND extraction by far. Moreover, the recall of ALL of proposed method is slightly lower than Study [Lu, Ma, Lu et al. (2016)] which use the method of improving recall on the premise of ensuring accuracy, the recall of ours is 47.83 and Study [Lu, Ma, Lu et al. (2016)] is 48.65. In general, the results show that our approach performed very well regarding F-score of ALL, which may be attributed to our EDPR algorithm filtering some noisy examples. Therefore, our method is shown to be effective, it can extract biomedical events well from the GE'13 test set.

Table 8: Performance comparison with other systems on GE'13 test set

Event class / System	SVT	BIND	REG	ALL
Ours	72.00/**82.85**/**77.04**	40.24/44.82/42.41	36.78/54.46/**43.91**	47.83/**63.20**/**54.45**
Study [Lu, Ma, Lu et al. (2016)]	74.12/77.56/75.80	39.34/44.11/41.59	**37.24**/45.74/41.05	**48.65**/56.24/52.17
EVEX	73.82/77.73/75.72	41.14/44.77/42.88	32.41/47.16/38.41	45.87/58.03/51.24
TEES 2.1	**74.52**/77.73/76.09	42.34/44.34/43.32	33.08/44.78/38.05	46.60/56.32/51.00
BIOSEM	67.71/86.90/76.11	**47.45**/**52.32**/**49.76**	28.19/49.06/35.80	42.47/62.83/50.68
NCBI	72.99/72.12/72.55	37.54/41.81/39.56	24.74/**55.61**/34.25	40.53/61.72/48.93

Performance is shown in recall/precision/F-score.

4 Discussion

This study has concentrated on error data detection to reduce noise from unlabeled corpus to improve the classifier performance. The proposed method involves analyzing the data and can be split into three steps: construction of mislabeled dataset, pair vector generation of error samples, and a sample identification strategy. Extensive experiments were conducted to evaluate the system's performance. These experimental results provide substantial evidence for the proposed method. Furthermore, pair vector representation of error samples has played a very important role, the representation and sample identification strategy in filter noisy samples performed very well. Overall, the proposed approach both achieves the desired effectiveness and improves the biomedical event extraction performance.

Despite the great advantages, the proposed method has also various disadvantages. First of all, events with minority classes samples are ignored when building mislabeled dataset. Too few samples are not conducive to the pattern generation of error samples. In this case, it may introduce more noise. Secondly, the effectiveness of extracting multi-argument events needs to be improved. And then, due to the limitation of GENIA corpus, the number of error samples of constructing mislabeled dataset is also limited. Finally, the validity of the proposed method is verified in GENIA corpus, but not in the latest biomedical literatures. In the future, we will expand the mislabeled dataset by adding samples of minority classes event and multi-argument event, test further by latest biomedical literature.

5 Conclusion

In this paper, we described a biomedical event extract approach based on error data detection using deep learning techniques. In the BioNLP Shared Task GENIA, the experiment results indicate that the proposed approach is competent in extract the biomedical event from biomedical literature. We compared the performance of the proposed approach with other systems and achieved a significant effect in biomedical event extract. Further research on biomedical complex event is necessary. The error detection for multi-parameter complex events can be used to improve our approach in future studies. And we will further study protein complexes through existing methods in the future.

Acknowledgement: This work was supported by the National Natural Science Foundation of China (No. 61672301), Jilin Provincial Science & Technology Development (20180101054JC), Science and Technology Innovation Guide Project of Inner Mongolia Autonomous Region of China (2017) and Talent Development Fund of Jilin Province (2018).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

Availability of Data and Materials: The datasets of BioNLP Shared Task GENIA 2011 can be download at: http://bionlp-st.dbcls.jp/GE/2011/downloads/. The datasets of BioNLP Shared Task GENIA 2013 can be download at: http://2013.bionlp-st.org/tasks. The unlabeled corpora including proteins annotations (bioconcepts2pubtator_offsets.gz) are downloaded from PubTator: ftp://ftp.ncbi.nlm.nih.gov/pub/lu/PubTator/. Word embedding is used published materials PubMed-and-PMC-w2v download at: http://evexdb.org/pmresources/vec-space-models/.

References

Antonov, A. V.; Dietmann, S.; Rodchenkov, I.; Mewes, H. W. (2009): PPI spider: a tool for the interpretation of proteomics data in the context of protein-protein interaction networks. *Proteomics*, vol. 9, no. 10, pp. 2740-2749.

Björne, J.; Ginter, F.; Salakoski, T. (2012): University of Turku in the bioNLP'11 shared task. *BMC Bioinformatics*, vol. 13 (Supplement 11), pp. 1-13.

Björne, J.; Heimonen, J.; Ginter, F.; Airola, A.; Pahikkala, T. et al. (2009): Extracting complex biological events with rich graph-based feature sets. *Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task*, pp. 10-18.

Hochreiter, S.; Schmidhuber, J. (1997): Long short-term memory. *Neural Computation*, vol. 9, no. 8, pp.1735-1780.

Hou, W. J.; Ceesay, B. (2015): Event extraction for gene regulation network using syntactic and semantic approaches. *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, vol. 9101, pp. 559-570.

Kai, H.; Landeghem, S. V.; Salakoski, T.; Peer, Y.; Ginter, F. (2013): EVEX in ST'13: Application of a large-scale text mining resource to event extraction and network construction. *Proceedings of the BioNLP Shared Task 2013 Workshop*, pp. 26-34.

Kim, J. D.; Ohta, T.; Pyysalo, S.; Kano, Y.; Tsujii, J. (2009): Overview of bioNLP'09 shared task on event extraction. *Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task*, pp. 1-9.

Kim, J. D.; Wang, Y.; Takagi, T.; Yonezawa, A. (2011): Overview of genia event task in BioNLP shared task 2011. *Proceedings of BioNLP Shared Task 2011 Workshop*, pp. 7-15.

Kim, J. D.; Wang, Y.; Yasunori, Y. (2013): The genia event extraction shared task, 2013 edition-overview. *Proceedings of the BioNLP Shared Task 2013 Workshop*, pp. 8-15.

Kolya, A. K.; Ekbal, A.; Bandyopadhyay, S. (2012): A hybrid approach for biomedical

event extraction. *Polibits*, vol. 46, no. 46, pp. 55-59.

Lee, P. C.; Ahmed, I.; Loriot, M. A.; Mulot, C.; Paul, K. C. et al. (2018): Smoking and Parkinson disease: evidence for gene-by-smoking interactions. *Neurology*, vol. 90, no. 7, pp. 583-592.

Lu, Y.; Ma, X. L.; Lu, Y. N.; Zhou, Y. X.; Pei, Z. L. (2016): A novel sample selection strategy for imbalanced data of biomedical event extraction with joint scoring mechanism. *Computational and Mathematical Methods in Medicine*, vol. 2, pp. 1-11.

Lu, Z. (2011): PubMed and beyond: a survey of web tools for searching biomedical literature. *Database*, vol. 2011, pp. 1-13.

Mcclosky, D.; Surdeanu, M.; Manning, C. D. (2011): Event extraction as dependency parsing for BioNLP 2011. *Proceedings of the BioNLP Shared Task 2011 Workshop*, pp. 41-45.

Mehryary, F.; Kaewphan, S.; Kai, H.; Ginter, F. (2016): Filtering large-scale event collections using a combination of supervised and unsupervised learning for event trigger classification. *Journal of Biomedical Semantics*, vol. 7, no. 1, pp. 1-13.

Miwa, M.; Saetre, R.; Kim, J. D.; Tsujii, J. (2010): Event extraction with complex event classification using rich features. *Journal of Bioinformatics and Computational Biology*, vol. 8, no. 1, pp. 131-146.

Munkhdalai, T.; Namsrai, O. E.; Ryu, K. H. (2015): Self-training in significance space of support vectors for imbalanced biomedical event data. *BMC Bioinformatics*, vol. 16 (Supplement 7), pp. 1-8.

Özgür, A.; Radev, D. R. (2009): Supervised classification for extracting biomedical events. *Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task*, pp. 111-114.

Pei, J.; Han, J.; Mortazavi-Asl, B.; Pinto, H.; Hsu, M. C. et al. (2002): Prefixspan: mining sequential patterns efficiently by prefix-projected pattern growth. *Proceedings of Icde Heidelberg Germany*, pp. 215-224.

Pyysalo, S.; Ginter, F.; Moen, H.; Salakoski, T.; Ananiadou, S. (2013): Distributional semantics resources for biomedical text processing. *Proceedings of the 5th Languages in Biology and Medicine Conference*, pp. 39-43.

Wang, J.; Xu, Q.; Lin, H.; Yang, Z.; Li, Y. (2013): Semi-supervised method for biomedical event extraction. *Proteome Science*, vol. 11 (Supplement 1), pp. 1-10.

Wei, C. H.; Kao, H. Y.; Lu, Z. Y. (2013): PubTator: a web-based text mining tool for assisting biocuration. *Nucleic Acids Research*, vol. 41, no. W1, pp. 518-522.

Xiao, L.; Bordes, A.; Grandvalet, Y. (2013): Biomedical event extraction by multi-class classification of pairs of text entities. *Proceedings of the BioNLP Shared Task 2013 Workshop*, pp. 45-49.

Xiong, Z.; Shen, Q.; Wang, Y; Zhu, C. (2018): Paragraph vector representation based on word to vector and CNN learning. *Computers, Materials & Continua*, vol. 55, no. 2, pp. 213-227.

Yamazaki, S. (2018): Relationships of changes in pharmacokinetic parameters of

substrate drugs in drug-drug interactions on metabolizing enzymes and transporters. *Journal of Clinical Pharmacology*, vol. 58, no. 8, pp. 1053-1060.

Yang, L.; Wei, F.; Li, S.; Ji, H.; Ming, Z. et al. (2015): A dependency-based neural network for relation classification. *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, vol. 2.

Zhou, C.; Sun, C.; Liu, Z.; Lau, F. C. M. (2015): A C-LSTM neural network for text classification. *Computer Science*, vol. 1, no. 4, pp. 39-44.

Zhou, D.; Zhong, D. (2015): A semi-supervised learning framework for biomedical event extraction based on hidden topics. *Artificial Intelligence in Medicine*, vol. 64, no. 1, pp. 51-58.