BMC Medical Informatics and Decision Making

RESEARCH Open Access

A multiple distributed representation method based on neural network for biomedical event extraction

Anran Wang, Jian Wang*, Hongfei Lin, Jianhai Zhang, Zhihao Yang and Kan Xu

From IEEE BIBM International Conference on Bioinformatics & Biomedicine (BIBM) 2016
Shenzhen, China. 15-18 December 2016

Abstract

Background: Biomedical event extraction is one of the most frontier domains in biomedical research. The two main subtasks of biomedical event extraction are trigger identification and arguments detection which can both be considered as classification problems. However, traditional state-of-the-art methods are based on support vector machine (SVM) with massive manually designed one-hot represented features, which require enormous work but lack semantic relation among words.

Methods: In this paper, we propose a multiple distributed representation method for biomedical event extraction. The method combines context consisting of dependency-based word embedding, and task-based features represented in a distributed way as the input of deep learning models to train deep learning models. Finally, we used softmax classifier to label the example candidates.

Results: The experimental results on Multi-Level Event Extraction (MLEE) corpus show higher F-scores of 77.97% in trigger identification and 58.31% in overall compared to the state-of-the-art SVM method.

Conclusions: Our distributed representation method for biomedical event extraction avoids the problems of semantic gap and dimension disaster from traditional one-hot representation methods. The promising results demonstrate that our proposed method is effective for biomedical event extraction.

Keywords: Biomedical event extraction, Distributed representation, Deep learning, Convolutional neural network

Background

With the exponentially increasing use of biomedical data, manually extracting useful information from massive biomedical text and associating them becomes imperative. As one of the information extraction areas, biomedical event extraction aims to extract more fine-grained and complex biomedical relations between entities such as biological molecules, cells, and tissues from texts and plays an important role in biomedical research [1].

A biomedical event is normally described by the type of event, a trigger and arguments. The goal of extraction is to detect the type of trigger and furthermore to detect the relation type between triggers with arguments. There are two common approaches on biomedical event extraction: rule-based and machines learning (ML)-based approaches. The rule-based approach builds up the model based on a dictionary and patterns generated from annotated events. Therefore, the model has a high precision on prediction but a low recall because of the poor generalization ability from stable rules. While ML-based approaches show a nice generalization ability on the recent Biomedical tasks [2]. The common pipeline for event extraction includes two subtasks: event trigger identification and event argument detection from documents that already contain Named Entity Recognition

* Correspondence: wangjian@dlut.edu.cn
School of Computer Science and Technology, Dalian University of Technology, Dalian, China

(NER). Each task can be regarded as a classification problem. The ML approaches, especially using SVM, have reached the best performance in recent BioNLP-ST tasks of the Genia Event Extraction (GE) [2–7].

The main advantage of the ML-based approach is that rich features can be extracted from text. However, the wide array of defined features may cause difficulties. Additional complex features engineering are needed when applying this model to different biomedical event extraction tasks. Moreover, the hand designed features are mostly one-hot features which are restricted by the problems of semantic gap and dimension disaster. On the other hand, the deep learning methods have achieved considerable progress after word2Vec [8], an effective tool of word embedding representation proposed on some natural language processing tasks such as named entity recognition [9] and sentence classification [10].

To decrease the complexity of feature engineering and detect underlying semantic relations among features automatically, we proposed a distributed representation method which contains not only word embedding information but also task-based distributed features like trigger type on deep learning models to realize event trigger identification and event argument detection to complete biomedical event extraction. Our method was applied on MLEE corpus [1]. This corpus was manually annotated by the guideline formalized in the BioNLP Shared Tasks on event extraction. Our method achieved a higher F-score on both trigger identification and event extraction compared to the baseline [3]. The results demonstrate that the proposed method is effective for biomedical event extraction.

Methods

As mentioned before, biomedical event extraction is to extract complex biomedical relations between biomedical entities. To simplify the task, the common work flow divides the task into some subtasks shown in Fig. 1. As NER is a mature field in NLP study, the corpus, which followed the guidelines of BioNLP Shared Tasks for biomedical event extraction, are already annotated with biomedical entities to decrease the complexity of this task. With the given entities and context information, triggers and arguments need to be detected to form biomedical events. After some post process with rules designed to filter out incorrect relations, biomedical event extraction is completed.

Our main work focuses on using the deep learning model with the distributed representation method on trigger and arguments detection to improve the accuracy of biomedical event extraction.

To let the model fully learn from training data, we describe each sample in two aspects. First, we utilize word embedding to form the basic representation of context. We parsed all available PubMed abstracts with Gdep [11] to get the contexts of dependency and fed the context from the dependency tree to word2vecf [12] to acquire word

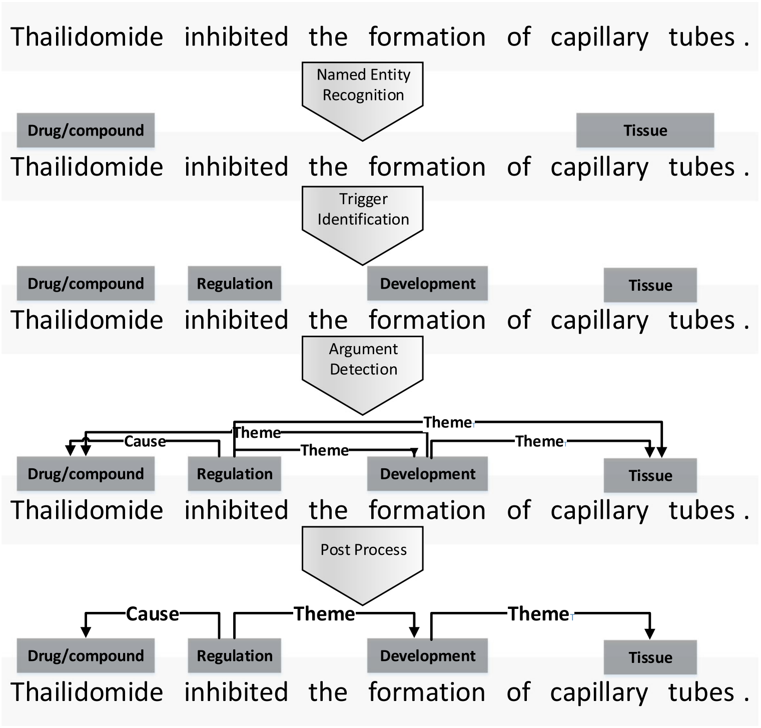
Thailidomide inhibited the formation of capillary tubes .
Named Entity Recognition
Drug/compound
Tissue
Thailidomide inhibited the formation of capillary tubes .
Trigger Identification
Drug/compound
Regulation
Development
Tissue
Thailidomide inhibited the formation of capillary tubes .
Argument Detection
Theme
Cause
Theme
Theme
Theme
Drug/compound
Regulation
Development
Tissue
Thailidomide inhibited the formation of capillary tubes .
Post Process
Cause
Theme
Theme
Drug/compound
Regulation
Development
Tissue
Thailidomide inhibited the formation of capillary tubes .

Fig. 1 The procedure of biomedical event extraction

embedding with stronger semantic relation. Second, we extended the representation with other task-based features' distributed representation like Part-of-speech (POS), distance which achieved positive results in previous works [3] to supplement sentence structure information or information that needs to be reinforced.

We discussed the distributed representation of two subtask problems and deep learning models in the follow sections in detail.

Event trigger identification

The main problem of trigger identification is to predict the trigger type of every word in a sentence. This can be represented as a classification problem shown as follow:

$$\mathcal{F}(\psi(\mathrm{w_i})) = \begin{cases} \mathcal{C} & \text{Positive}, \mathcal{C} \text{ is the type id of trigger} \\ -1 & \text{Negative} \end{cases} \tag{1}$$

Where $\psi(\mathrm{w_i})$ denotes distributed representation of each word, and $\mathcal{F}$ is a classification function. The model we have chosen is the deep neural network, the framework is shown as Fig. 2.

Multiple distributed representation construction

A sentence might contain more than one event, but most events will be contained in a single sentence. To decrease the complexity of the model, we divided corpus into sentences and built feature representations from sentence information. Inspired by the language model, we drew the basic picture of trigger candidates by their context and expanded details by their POS. Due to the deep relevance between triggers and entities (participate candidates), we utilized the distance between them to describe the possibility of the candidate to be a trigger. Moreover, we added sentence topic probability to discriminate sentences.

- Context representation

As mentioned before, we adopted the context words between candidate triggers to predict trigger type. Given a sentence $\mathrm{S = \ldots w_{i-2} w_{i-1} T_i w_{i+1} w_{i+2} \ldots}$, where $\mathrm{T_i}$- denotes trigger to be predicted, and assuming the window size ($\mathrm{d_{win}}$) is 2, therefore words $\mathrm{w_{i-2}}$, $\mathrm{w_{i-1}}$, $\mathrm{w_{i+1}}$, $\mathrm{w_{i+2}}$ are what we will adopt.

First, we found words around the trigger to be predicted from a dictionary of training set according to $\mathrm{d_{win}}$, then we initialized these words' vectors according to the dependency-based word embedding table. If the window is out of the edge of the sentence or can't be found in the word embedding table, then we initialize the embedding in a random way. Finally, we concatenated them as follows:

$$\psi_{\mathrm{con}}(\mathrm{T_i}) = \Big[\langle \mathrm{W} \rangle_{[\mathrm{w_{i-d_{win}}}]} \cdot \cdots \cdot \langle \mathrm{W} \rangle_{[\mathrm{w_{i-1}}]} \cdot \langle \mathrm{W} \rangle_{[\mathrm{T_i}]} \cdot \langle \mathrm{W} \rangle_{[\mathrm{w_{i+1}}]} \cdots \langle \mathrm{W} \rangle_{[\mathrm{w_{i+d_{win}}}]}\Big] \tag{2}$$

Where $\mathrm{W} \in \mathcal{R}^{\mathrm{dim} \times |\mathcal{D}|}$ represents the dependency-based word embedding table, dim is the dimensionality of word embedding we trained in the previous work, and $|\mathcal{D}|$ is the size of the dictionary.

- Topic representation

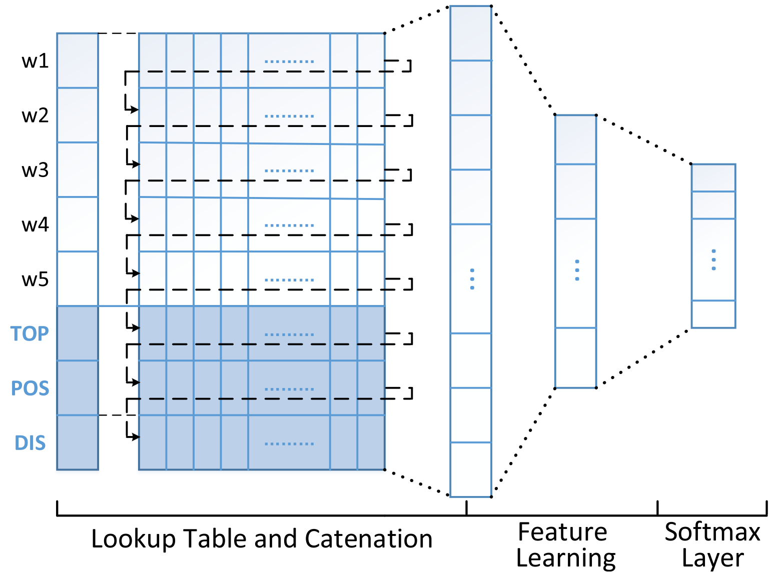

Fig. 2 The model of trigger identification

Ambiguity is a problem that can't be neglected in word classification tasks. A word can express various meanings according to the specific context. Therefore, disambiguation plays an important role in word classification tasks including trigger identification.

Here we added the topic feature to represent sentence topic information of each trigger in the sentence. We used a Latent Dirichlet Allocation (LDA) [13] tool provided by Gensim, a python library, to acquire the topic distribution of every word in a sentence (the probability of a word belonging to one topic), and then multiply the topic probability of every word in one sentence to get the approximate sentence topic distribution [14]. The sentence topic feature is computed by the following equation:

$$\psi_{\mathrm{top}}(\mathrm{S}) = \prod\nolimits_{\mathrm{j=i-d_{win}}}^{\mathrm{i+d_{win}}} \langle \mathrm{W_{top}} \rangle_{[\mathrm{w_j}]} \tag{3}$$

Where $\langle \mathrm{W_{top}} \rangle \in \mathcal{R}^{\mathrm{d_{top}} \times |\mathrm{D}|}$ is words topic distribution, $\mathrm{d_{top}}$ represents the total number of topics, and $|\mathrm{D}|$ denotes the size of the dictionary.

- POS representation

The POS of triggers are mostly related to verbs which makes them a considerable role in trigger identification. The distributed representation of POS is shown as follow:

$$\psi_{\mathrm{pos}}(\mathrm{T_i}) = \langle \mathrm{W_{pos}} \rangle_{[\mathrm{pos_{T_i}}]} \tag{4}$$

Where $\langle \mathrm{W_{pos}} \rangle \in \mathcal{R}^{\mathrm{d_{pos}} \times |\mathcal{D}_{\mathrm{pos}}|}$ represent POS vector table, $|\mathcal{D}_{\mathrm{pos}}|$ is the number of different POS types in the training set, and $\mathrm{d_{pos}}$ is the dimensionality of the POS vector. $\mathrm{W_{pos}}$ is initialized at random and updated while training.

- Distance representation

Triggers have a deep relevance with the entities. The longer the distance between the word and the entities, the lower the possibility it can be a trigger. Based on our previous work [15], we adopt the distance between the trigger and entities in the dependency tree, and the distance is shown as follows:

$$\psi_{\mathrm{dis}}(\mathrm{T_iE_j}) = \langle \mathrm{W_{dis}} \rangle_{[\mathrm{dis}(\mathrm{T_{i,}, E_j})]} \tag{5}$$

Where $\mathrm{dis}(\mathrm{T_i, E_j})$ represents the distance of the candidate trigger to the closest entities, and $\langle \mathrm{W_{dis}} \rangle \in \mathcal{R}^{\mathrm{d_{dis}} \times |\mathcal{D}_{\mathrm{dis}}|}$ is the distance table. $|\mathcal{D}_{\mathrm{dis}}|$ is the number of distance possibilities and $\mathrm{d_{dis}}$ is the dimensionality of the distance vector. $\mathrm{W_{dis}}$ is initialized at random and updated while training.

Finally, the concatenation of these four feature representations formed the distributed representation of candidate triggers to be the input layer fed to the deep learning model.

$$\psi(\mathrm{T_i}) = \psi_{\mathrm{con}}(\mathrm{T_i}) \cdot \psi_{\mathrm{top}}(\mathrm{S}) \cdot \psi_{\mathrm{pos}}(\mathrm{T_i}) \cdot \psi_{\mathrm{dis}}(\mathrm{T_iE_j}) \tag{6}$$

Where · denotes concatenation.

Table 1 The setting of parameters in neural network for event trigger identification

	Word embedding dimension	Hidden layers	Hidden-layer notes	Batch	Dropout Rate
Value	100	3	1000	512	0.2

Deep learning model

We fed $\psi(T_i)$ to the deep learning model provided by Theano [16] and sped up computation by running on the GPU. To optimize the learning model, we utilized Adadelta [17] to adjust the learning rate automatically and used Dropout [18] to prevent the model from over fitting. The parameters were set as in Table 1.

Event argument detection

To extract the event, we need to detect event arguments after trigger identification. This step is for examining whether entities have any relation with the triggers and detecting the type of relation. In the end, an event can be generated from the composition of a trigger and its event arguments. The process can be represented as a classification problem and modeled as follows:

$$\mathcal{F}(\psi(T_i(E_j|T_k))) = \begin{cases} \mathcal{C} & \text{Positive}, \mathcal{C} \text{ is the type id} \\ & \text{of event argument} \\ -1 & \text{Negative} \end{cases} \tag{7}$$

Where ψ denotes the feature extraction function, $\mathrm{T_i(E_j|\,T_k)}$ denotes the relation of entity $\mathrm{E_j}$ with trigger $\mathrm{T_k}$ ($\mathrm{E_j}$ can be an argument or a trigger).

Some previous works showed that convolutional neural network (CNN) has significant effectiveness in the modeling of sentences [10]. The ability of gaining feature maps by different filters helps to exploit rich and deep features. Also CNN can unify the length from various distances between triggers and participate candidates. Therefore, we employed CNN to model the sentence based on the dependency path between the trigger and candidates to detect trigger-argument or trigger-trigger relation. The framework is shown as Fig. 3.

Table 2 The parameters of convolutional neural network for event argument detection

	Word embedding dimension	Filters	Hidden-layer notes	Batch	Dropout
Value	50	[3, 5, 7]	1000	128	0.2

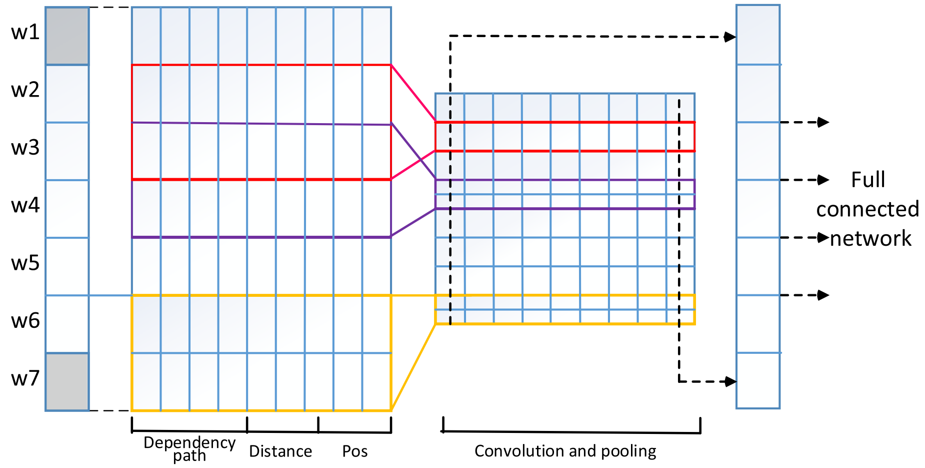

Fig. 3 The model of argument detection based on convolutional neural network

Multiple distributed representation construction

To detect the relation of triggers and arguments, we utilized the words between them as the fundamental representation of their relations. We also added POS and distance to describe words on the dependency path more sufficiently. In addition, we used the type of trigger and argument to express their type of relation.

- Dependency-path representation

Traditional methods in relation classification tasks use words before two entities to represent their relation. However due to the high frequency of long and complex sentences in biomedical corpus, it is hard to detect semantic relation between trigger and arguments. To solve that, we adopted the word on the dependency-path between the trigger and candidates to represent the logical relation of them.

After parsing a sentence, we get the dependency path $S = w_1w_2w_3w_4w_5...w_n$ between trigger and entities. We then convert the sequence to a matrix according to the word embedding:

$$\psi_{con}(S) = \begin{bmatrix} \langle W\rangle_{[w_1]}{}^{1} & \cdots & \langle W\rangle_{[w_n]}{}^{1} \\ \vdots & \ddots & \vdots \\ \langle W\rangle_{[w_1]}{}^{d} & \cdots & \langle W\rangle_{[w_n]}{}^{d} \end{bmatrix} \quad (8)$$

Where $W \in \mathcal{R}^{d\times|\mathcal{D}|}$ represents word embedding table as mentioned before.

- POS representation

Different from POS representation in trigger identification, we utilized POS from words on the dependency path:

$$\psi_{pos}(S) = \begin{bmatrix} \langle W_{pos}\rangle_{[pos(w_1)]}{}^{1} \cdots \langle W_{pos}\rangle_{[pos(w_n)]}{}^{1} \\ \vdots \quad \ddots \quad \vdots \\ \langle W_{pos}\rangle_{[pos(w_1)]}{}^{d} \cdots \langle W_{pos}\rangle_{[pos(w_n)]}{}^{d} \end{bmatrix} \quad (9)$$

Where $\langle W_{pos}\rangle \in \mathcal{R}^{d_{pos}\times|\mathcal{D}_{pos}|}$ represents POS embedding table, which is initialized at random and updated while training.

- Distance representation

There are two distances we measured to capture the relative distance. One is the distance from word w_i in S to the first entity (or trigger), the other one is from w_i to the second entity (or trigger) shown as follows respectively:

Table 3 The statistics of MLEE dataset

Data	Train	Validation	Test	All
Document	131	44	87	262
Sentence	1271	457	880	2608
Word	27,875	9610	19,103	56,588
Entity	4147	1431	2713	8291
Event	3296	1175	2206	6677

Table 4 The comparison of results on trigger identification in MLEE

Method	F-score (%)	Precision (%)	Recall (%)
Proposed method	77.97	80.92	75.23
Zhou [3]	76.89	72.17	82.26
Pyysalo [1]	75.84	70.79	81.69

Table 5 The comparison of results on events extraction in MLEE

Method	F-score (%)	Precision (%)	Recall (%)
Proposed method	58.31	60.56	56.23
Zhou [3]	57.41	55.76	59.16
Pyysalo [1]	55.20	62.28	49.56

$$\psi_{\mathrm{dis\_e1}}(\mathrm{S}) = \begin{bmatrix} \langle \mathrm{W_{dis}} \rangle_{[\mathrm{dis}(\mathrm{w_1},\mathrm{e_1})]}{}^{1} & \cdots & \langle \mathrm{W_{dis}} \rangle_{[\mathrm{dis}(\mathrm{w_n},\mathrm{e_1})]}{}^{1} \\ \vdots & \ddots & \vdots \\ \langle \mathrm{W_{dis}} \rangle_{[\mathrm{dis}(\mathrm{w_1},\mathrm{e_1})]}{}^{\mathrm{d_{dis}}} & \cdots & \langle \mathrm{W_{dis}} \rangle_{[\mathrm{dis}(\mathrm{w_n},\mathrm{e_1})]}{}^{\mathrm{d_{dis}}} \end{bmatrix} \tag{10}$$

$$\psi_{dis\_e2}(S) = \begin{bmatrix} \langle W_{dis} \rangle_{[dis(w_1,e_2)]}{}^{1} & \cdots & \langle W_{dis} \rangle_{[dis(w_n,e_2)]}{}^{1} \\ \vdots & \ddots & \vdots \\ \langle W_{dis} \rangle_{[dis(w_1,e_2)]}{}^{d_{dis}} & \cdots & \langle W_{dis} \rangle_{[dis(w_n,e_2)]}{}^{d_{dis}} \end{bmatrix} \tag{11}$$

Where, $\langle \mathrm{W_{dis}} \rangle \in \mathcal{R}^{\mathrm{d_{dis}} \times |\mathcal{D}_{\mathrm{dis}}|}$ represents the distance embedding table, which is initialized at random and updated while training.

- Type representation

According to the domain knowledge of the corpus, we know that there are many constant type combinations such as entity "Cell" that are more likely to have a relation to the event type "Cell proliferation". To describe the probability of the relation two entities have, we combine the type of triggers or entities with dependency-path word representation to describe the sentence feature.

$$\psi_{type}(S) = \begin{bmatrix} \langle W_{type} \rangle_{[e_1]}{}^{1} & \langle W_{type} \rangle_{[e_2]}{}^{1} \\ \vdots & \vdots \\ \langle W type \rangle_{[e_1]}{}^{d_{type}} & \langle W_{type} \rangle_{[e_2]}{}^{d_{type}} \end{bmatrix} \tag{12}$$

Where $\langle \mathrm{W_{type}} \rangle \in \mathcal{R}^{\mathrm{d_{type}} \times |\mathcal{D}_{\mathrm{type}}|}$ represents the entities type table, which is initialized at random and updated while training as well. $|\mathcal{D}_{\mathrm{type}}|$ is the total type number, and $\mathrm{d_{type}}$ is the dimension of each type which should be the same as word embedding. For the type and dependency words all represent sentence feature and we concatenate them as one feature.

Table 6 The comparison of results with different features on trigger identification

Method	F-score (%)	Precision (%)	Recall (%)
context	75.51	78.83	72.47
context +topic	75.96	79.88	72.42
context +POS	75.98	79.84	72.47
context +distance	77.27	80.60	74.18
context + topic+ POS + distance	77.97	80.92	75.23

Table 7 The comparison of results with different features on events extraction

Method	F-score (%)	Precision (%)	Recall (%)
context	55.60	58.34	53.10
context +POS	56.24	59.28	53.49
context +distance	57.68	60.04	55.49
context +type	57.40	59.21	55.69
context +POS + distance + type	58.31	60.56	56.23

Finally, we map these four distributed representations as the description of sentences and employ CNN to build the classification model.

$$\psi(S) = \begin{bmatrix} \psi con(S) \cdot \psi_{type}(S) \\ \psi pos(S) \\ \psi_{dis_e 1}(S) \\ \psi_{dis_e 2}(S) \end{bmatrix} \tag{13}$$

Convolutional neural network

The main distinguished layers of CNN from other neural network are convolutional layer and pooling layer. The filters, provided in the convolutional layer, have a small receptive field and extend through all input volume with shared parameters to learn the various levels of features from the input layer. We set three filters based on our sentence length of corpus to extract rich features. The pooling layer helps to progressively reduce the size of the representation to reduce the amount of parameters and computation in the network. We adopted max-pooling to extract the most valuable feature representation. The full connected network is similar to the model mentioned above. The parameters were set as in Table 2.

Results and discussion

Datasets and evaluation metrics

We evaluated our proposed method on MLEE corpus, which spanned all levels of biomedical organization and covered 19 types of triggers. The details of the dataset is shown in Table 3.

In an event extraction task, there are three parts that need to be equal when identifying if two events are equal: event type, event trigger, and argument. In this

Table 8 The comparison with the best performance of event extraction results in BioNLP-ST2009 dataset

Method	F-score (%)	Precision (%)	Recall (%)
Proposed method	59.94	64.34	56.10
EventMine [4]	58.81	63.17	55.00
UMASS [5]	58.70	–	–

paper we adopted two methods from BioNLP-ST task to evaluate identity, the approximate span matching method to evaluate if the predicted trigger is equal to the annotated trigger and the approximate recursive matching method. This was to judge if two events are equal. Further, we measured the performance of our proposed method by F-score, precision and recall.

Experimental results

To evaluate our proposed method in this paper, we compare the performance of the two baselines constructed by Zhou [3], which are state-of-the-art methods based on the MLEE corpus as well as using SVM. Then we tested different feature combinations to detect the contribution from different features. Furthermore, we applied our method on datasets of GE task in BioNLP-ST2009, 2011, 2013 as well to inspect the generalization ability of the method.

- Performance comparison with other methods

We compared our method with the two state-of-the-art methods on trigger identification and event extraction in Tables 4 and 5. Based on the results, we achieved a better F-score performance on both trigger identification and event extraction. The result indicates that the proposed distributed representation, including dependent context which was formed by word embedding and task-based features on the deep learning methods, works well on biomedical event extraction tasks compared to the baseline approach using SVM with manually complex features. The lower recall might be caused by, insufficiency of the training data from event types appearing too seldom to be learned, or different post process rules.

- Feature analysis

The distributed representation of NLP tasks are mostly based on context information. In Tables 6 and 7, we listed the feature combination performance on F-score, precision, and recall to analyze each feature's contribution.

Based on the results from Tables 6 and 7, we can find that distance is a significant feature for trigger identification, beside context, which verifies the deep relevance between entities (arguments) and triggers. Also domain knowledge "type" in Table 7 is shown to have certain contributions to the performance.

All the extra features we added on context are the main features in SVM, which indicates that manually extracted features still contribute to deep learning models either by extending features not learned by the model or by reinforcing the features already learned by model.

Table 9 The comparison with the best performance of event extraction results inBioNLP-ST2011 dataset

Method	F-score (%)	Precision (%)	Recall (%)
FAUST [2]	55.90	–	–
Proposed method	55.20	58.33	52.38
UMASS [2]	54.80	–	–

- Generalization assessment

To examine our proposed method's generalization ability on other biomedical corpus, we applied our method on the latest BioNLP event extraction tasks as well. The baselines we chose are distinguished event extraction tools that emerged in recent years and achieved outstanding performance in the tasks respectively. EvenMine [4] and UMASS [5] yield better results than the methods in the evaluation of BioNLP-ST'09. FAUST [2] is a model based on UMASS and achieved the best performance in BioNLP-ST'11. EVEX [6]. This method is ranked first in BioNLP-ST'13 and uses the output of the TEES system as features of confidence scores from other models and added extra sentence structure information features for training on SVM.

Tables 8, 9, 10 demonstrate that our proposed deep learning model based on basic pipeline of biomedical event extraction already achieved comparative results to all the SVM methods on all three datasets. From previous experience, the more complex a SVM model becomes the higher its performance. The latest SVM methods focus more on model combination and added task-based rules to improve the performance on specific evaluation task dataset. On the other hand, the amount of training data decide if a deep learning model can learn fully. However, our proposed method reached the same level of the best performance of evaluation tasks only with basic work flow and few task-based features which indicates the operation simplicity and generalization ability of our method.

Conclusion and future work

In this paper, we have presented a multiple distributed representation method which combines dependent context formed by word embedding with task-based features from biomedical text and fed it to deep learning models to achieve biomedical event extraction. This method

Table 10 The comparison with the best performance of event extraction results in BioNLP-ST2013 dataset

Method	F-score (%)	Precision (%)	Recall (%)
EVEX [6]	50.97	58.03	45.44
TEES-2.1 [7]	50.74	56.32	46.17
Proposed method	50.12	55.67	45.58

avoids the problems of semantic gap and dimension disaster from traditional one-hot representation methods and achieved a promising result on several datasets using the basic pipeline of event extraction with few extra manually extracted features instead of complex feature engineering. In particular, the distributed representation can be extended as the supplement of word embedding depending on the needs of specific tasks. Besides the semantic information from word embedding, the model can extend task-based representation according to the domain knowledge, which makes the model more flexible for different topic based biomedical event extraction tasks.

In the future we plan to modify our deep learning models with attention method. Also we can apply our work on a Recurrent Neural network to learn sequence features and then combine them with the CNN model to enrich the features of the learning model.

Acknowledgements
Authors would like to thank the National Key Research Development Program of China and the Natural Science Foundation of China for funding our research.

Funding
The research and the publication are supported by the National Key Research Development Program of China (No. 2016YFB1001103), the Natural Science Foundation of China (61,572,098, 61,572,102 and 61,602,078).

Availability of data and materials
The data (MLEE corpus) used in our experiment can be downloaded in http://nactem.ac.uk/MLEE. The data is available for public and free to use.

About this supplement
This article has been published as part of *BMC Medical Informatics and Decision Making* Volume 17 Supplement 3, 2017: Selected articles from the IEEE BIBM International Conference on Bioinformatics & Biomedicine (BIBM) 2016: medical informatics and decision making. The full contents of the supplement are available online at https://bmcmedinformdecismak.biomedcentral.com/articles/supplements/volume-17-supplement-3.

Authors' contributions
AW & JW carried out the biomedical natural language studies and the biomedical event extraction study. AW & JZ jointly developed the model. AW wrote the first draft. All authors read and approved the final manuscript.

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing financial interests.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Published: 20 December 2017

References

1. Pyysalo S, Ohta T, Miwa M, et al. Event extraction across multiple levels of biological organization[J]. Bioinformatics. 2012;28(18):i575.
2. Riedel S, Mcclosky D, Surdeanu M, et al. Model combination for event extraction in BioNLP 2011[C]. Proceedings of BioNLP Shared Task 2011 Workshop, 2011:51-55.
3. Zhou D, Zhong D. A semi-supervised learning framework for biomedical event extraction based on hidden topics.[J]. Artif Intell Med. 2015;64(1):51–8.
4. Miwa M, Thompson P, Ananiadou S. Boosting automatic event extraction from the literature using domain adaptation and coreference resolution[J]. Bioinformatics. 2012;28(13):1759–65.
5. Riedel S, Mccallum A. Fast and robust joint models for biomedical event extraction[C]. Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing. 2011:1-12.
6. Kai H, Landeghem SV, Salakoski T, et al. EVEX in ST'13: application of a large-scale text mining resource to event extraction and network construction[C]. Proceedings of the BioNLP Shared Task 2013 Workshop. 2013:26–34.
7. Björne J, Salakoski T. TEES 2.1: automated annotation scheme learning in the BioNLP 2013 shared task[C]. Proceedings of the BioNLP Shared Task 2013 Workshop. 2013:16–25.
8. Mikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality[J]. Advances in Neural Information Processing Systems. 2013(26): 3111–19.
9. Tang B, Cao H, Wang X, et al. Evaluating word representation features in biomedical named entity recognition tasks.[J]. Biomed Res Int. 2014;2014(2):240403.
10. Ma M, Huang L, Zhou B, Xiang B. Dependency-based Convolutional neural networks for sentence embedding[C]. Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers); 2015: p174–179.
11. Sagae K, Tsujii JI. Dependency parsing and domain adaptation with LR models and parser ensembles[C]. Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL. 2007:1044-50.
12. Levy O, Goldberg Y. Dependency-based word embeddings[C]. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). 2014:302–308.
13. Blei DM, Ng AY, Jordan MI. Latent dirichlet allocation[J]. J Mach Learn Res. 2003;3:993–1022.
14. Mikolov T, Zweig G. Context dependent recurrent neural network language model[C]. Proceedings of the Spoken Language Technology Workshop, IEEE. 2012:234-239.
15. Wang J, Zhang J, An Y, et al. Biomedical event trigger detection by dependency-based word embedding[J]. BMC Medical Genomics. 2016, vol 9,Suppl 2: 45.
16. Bergstra J, Breuleux O, Bastien F, et al. Theano: a CPU and GPU math compiler in python[C]. Proceedings of the 9th Python in Science Conf. 2010:1–7.
17. Zeiler M D. ADADELTA: an adaptive learning rate method[J]. arXiv preprint arXiv.2012:1212.5701.
18. Srivastava N, Hinton G, Krizhevsky A, et al. Dropout: a simple way to prevent neural networks from overfitting[J]. J Mach Learn Res. 2014;15(1):1929–58.