Using Mention Segmentation to Improve Event Detection with Multi-head Attention

Jiali Chen, Yu Hong, Jingli Zhang, and Jianmin Yao
School of Computer Science and Technology, Soochow University
Suzhou, China
{*ivycjl94, tianxianer, jlzhang05*}*@gmail.com, jyao@suda.edu.cn*

***Abstract*—Sentence-level event detection (ED) is a task of detecting words that describe specific types of events, including the subtasks of trigger word identification and event type classification. Previous work straightforwardly inputs a sentence into neural classification models and analyzes deep semantics of words in the sentence one by one. Relying on the semantics, probabilities of event classes can be predicted for each word, including the carefully defined ACE event classes and a "N/A" class(i.e., non-trigger word). The models achieve remarkable successes nowadays. However, our findings show that a natural sentence may posses more than one trigger word and thus entail different types of events. In particular, the closely related information of each event only lies in a unique sentence segment but has nothing to do with other segments. In order to reduce negative influences from noises in other segments, we propose to perform semantics learning for event detection only in the scope of segment instead of the whole sentence. Accordingly, we develop a novel ED method which integrates sentence segmentation into the neural event classification architecture. Bidirectional Long Short-Term Memory (Bi-LSTM) with multi-head attention is used as the classification model. Sentence segmentation is boiled down to a sequence labeling problem, where BERT is used. We combine embeddings, and use them as the input of the neural classification model. The experimental results show that the performance of our method reaches 76.8% and 74.2% F_1-scores for trigger identification and event type classification, which outperforms the state-of-the-art.**

***Keywords*-Event Detection; Mention Segmentation; Multi-head Attention;**

I. Introduction

Event detection (ED) is a crucial subtask of event extraction, which aims to identify triggers in a target sentence and assigns an event type for each trigger. For example, considering the sentence 1) on the ACE 2005[1], where "*pay*" is the trigger, and an ED system is expected to predict the "*Fine*" event triggered by "*pay*".

1) *He also brought a check from Campbell to* **pay** *the fines and fees.*

Recently, neural networks have been widely used in ED [1], which can capture semantic information. However, they can't ignore the interference of redundant words. If the model can concentrate more on event-related content, the ED performance will improve. For example, in sentence 2), "*discussions*" triggers the "*Meet*" event and the underlined part is an event mention. The content before the underline in the sentence is a supplement to the underlined content, rather than the principal content of the event. Thus, the content before the underline is redundant information for ED. If we consider more about the event-related information such as the underlined content, we would have more confidence in predicting the "*Meet*" event successfully. Manual annotation of event mention is time- and resource-expensive. Therefore, we propose a method for extracting event-related information. We adopt the idea of pointer networks [2] to get two pointers (indexes). One pointer points to the beginning of the event mention in the sentence, and the other pointer points to the end of the event mention. Then, the chunk divided from the sentence by two pointers is the used event mention in this paper. Finally, we use mention segmentation to improve ED performance.

2) *"It was useful to get it all out on the table and see where we go from here," he said, referring to the April 23 to 25* ***discussions*** *in Beijing.*

In addition, we note that the contribution degrees of all words in the sentence are different. For example, in sentence 1), "*pay*" provides more crucial clues than other words for ED and should be paid more attention. Fortunately, we find that multi-head attention mechanism [3] can compute the weights of different words in the sentence. And, it can help the Bidirectional Long Short-Term Memory (Bi-LSTM) [4] network concentrate on the important words in the sentence. Therefore, we add multi-head attention mechanism to the Bi-LSTM network with event mention for the ED task.

In summary, the contributions of this paper are as follows:

- We find that event mention is effective, and we also analyze the effects of event mention and use a model to predict event mention.
- We propose a novel method for improving ED that can exploit event mention via Bi-LSTM based on the multi-head attention. Additionally, we analyze the effects of multi-head attention.
- The experimental results demonstrate that our proposed method significantly outperforms the current state-of-the-art performance on the widely used ACE 2005 dataset.

II. Related Work

Event detection which aims to identify triggers and classify event types has achieved considerable results.

[1]https://catalog.ldc.upenn.edu/ldc2006t06

978-1-7281-5014-7/19/$31.00

Gupta et al. [5] propose cross-event inference to alleviate the problem of unknown time argument. Grishman and Ralph [6] achieve sentence-level event type classification with document-level information. Hong et al. [7] use cross-entity inference to achieve sentence-level trigger. Li et al. [8] incorporate global and local features via structured prediction. Liu et al. [9] propose a global inference method to implement event detection. The above methods are all feature-based methods.

At present, most studies have applied neural networks. Nguyen et al. [10] use Convolutional Neural Network (CNN) for event detection. Chen et al. [11] propose a dynamic multi-pooling convolutional neural network. Nguyen et al. [12] use a Bidirectional Recurrent Neural Network (Bi-RNN) to extract event triggers and arguments jointly. Duan et al. [1] exploit document-level information via Recurrent Neural Network (RNN). Feng et al. [13] propose a hybrid neural network. Chen et al. [14] propose a hierarchical and bias tagging networks with gated multi-level attention mechanisms. Zhao et al. [15] propose a novel document embedding enhanced Bi-RNN method. Hong et al. [16] propose a self-regulated learning method by exploiting a generative adversarial network to generate spurious features.

BERT [17] has emerged as an increasingly popular pre-trained language representation model for the tasks of natural language processing. For example, Xu et al. [18] use BERT post-training for review reading comprehension and aspect-based sentiment analysis. We follow the work to fine-tune BERT for extracting event mention. In addition, guided by the transformers [3] of the BERT encoder model, we note that the multi-head attention mechanism can capture important words in the sentence.

III. Task Description

This paper focuses on the ED task defined in ACE evaluation [19]. We will briefly introduce related terms for the ED task in this section.

- **Entity**: An object or collection of objects in one of the semantic categories, such as human, object, location, etc;
- **Entity Mention**: A phrase for a specific type of entity;
- **Event Trigger**: A main word that expresses the occurrence of an event (often a noun or phrase) and consists of a single word or phrase;
- **Event Argument**: The participant of an event, which is the important part of the event, involving entities, time, and values;
- **Event Mention**: A phrase or sentence which includes trigger words and event arguments.

An event involves a specific type of event and one or more participants associated with the event. The ED task aims to identify triggers and classify event types. The ACE 2005 evaluation defines 8 event types and 33 event subtypes. Following Li et al. [8], we only focus on 33 event subtypes.

IV. Methodology

Following Nguyen and Grishman [12], event detection is considered as a multi-class classification problem. The aim of the ED task is to predict whether the token in a given sentence can trigger a specific type of event. In this section, we will describe the event segmentation and the multi-head attention-based Bi-LSTM.

A. Mention Segmentation

In this paper, we propose a method that extracts event mention to capture the most related words about events in the sentence and reduce the interference of redundant words.

With the development of deep learning, we adopt the end-to-end model to extract event mention by fine-tuning the pre-trained BERT. BERT [17] is a pre-trained language representation model which can break down records of multiple tasks with one additional fully-connected layer. Thus, we select fine-tuning the pre-trained BERT model[2] for extracting event mention. Given a sentence with n words, and we formulate the input of BERT as $S = \{[CLS], s_1, s_2, ..., s_i, ..., s_n, [SEP]\}$, where $[CLS]$ is a dummy token and $[SEP]$ marks the end of the sentence. Firstly, we obtain the hidden representation after BERT as $h_s = BERT(S)$. Then, we add two separate linear layers to the hidden representation followed by a softmax function to get two pointers (indexes). The two pointers can indicate the start and end position of the target chunk. The specific calculation formula is as follows:

$$L_{start,end} = softmax(W_{1,2}h_s + b_{1,2}) \tag{1}$$

where $W_{1,2}$ are two separate weight matrices and $b_{1,2}$ are two separate bias terms. We utilize the averaged cross-entropy on the two pointers as to the loss function and minimize it via Adam optimizer:

$$L_{BERT} = -\frac{\sum T(start) log L_{start} + \sum T(end) log L_{end}}{2} \tag{2}$$

where $T(start)$ and $T(end)$ are one-hot vectors representing the ground-truth pointers. Finally, the chunk divided from the sentence by two predicted pointers is regarded as the predicted event mention.

B. Multi-head attention-based Bi-LSTM

Figure 1 demonstrates the architecture of multi-head attention-based Bi-LSTM for ED. The model consists of the following components: (i) Embedding Layer; (ii) Multi-head Attention Mechanism; (iii) Bi-LSTM; (iv) Output; (v) Training.

1) Embedding Layer: Following Liu et al. [20], we take each token in the target sentence as the input of the network and transform them into a real-valued vector by looking up embedding tables.

- **Word Embedding** We utilize the Skip-gram model [21] to learn word embedding on the NYT corpus[3]. We present it as: $W = \{w_1, w_2, ..., w_i, ..., w_n\}$,

[2]https://github.com/google-research/bert
[3]https://catalog.ldc.upenn.edu/LDC2008T19

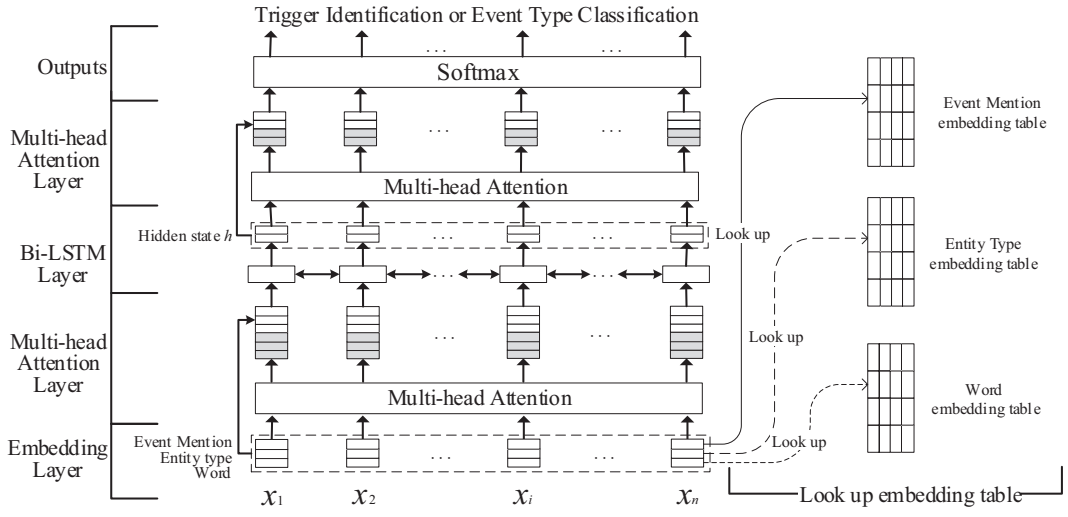

Figure 1. The architecture overview of multi-head attention-based Bi-LSTM with event mention

where w represents the hidden semantic properties of a token which is obtained by looking up a pre-trained word embedding table.

- **Entity Type Embedding** The representation of entity type in this sentence is: $E = \{0, ...e_i, ..., e_j, 0, ...\}$, where "0" represents that the current word is not an entity, and e_i or e_j indicates that the ith or jth word is a specific type of entity. Following Hong et al. [16], we look up entity type embedding table which is randomly initialized embedding vector to get the entity type embedding W_e.
- **Event Mention Embedding** We present the event mention label as: $M = \{0, ..., 1, ..., 1, 0, ...\}$, where "1" indicates event mention in a sentence and "0" indicates other words of the sentence. Similarly, we encode the event mention as a fixed-dimensional real-valued vector W_{men} by looking up randomly initialized event mention embedding table.

Finally, we concatenate the three embeddings into the final representation denoted as $X = \{W, W_e, W_{men}\}$, where $X \in \mathbb{R}^{n \times m}$, n is the length of the sentence and m is the dimensionality of each token.

2) Multi-head Attention Mechanism: For paying more attention to keywords in a given sentence and reducing the interference of meaningless words, we exploit the multi-head attention mechanism [3] before and after Bi-LSTM respectively. In this section, we will introduce the details of multi-head attention.

The structure of multi-head attention is shown in Figure 2, the three inputs of multi-head attention are "Query", "Key", "Value", which are denoted as Q, K, V. Since we adopt the self-attention in one sentence, we set "$Q = K = V = X$". Firstly, we add three separate linear layers to the Q, K, V, and denote the outputs as $\tilde{Q}$, $\tilde{K}$, $\tilde{V} \in \mathbb{R}^{n \times m}$:

$$\begin{bmatrix} \tilde{Q} \\ \tilde{K} \\ \tilde{V} \end{bmatrix} = W \begin{bmatrix} Q \\ K \\ V \end{bmatrix} + b \tag{3}$$

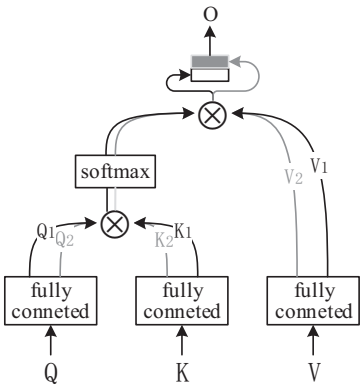

Figure 2. Multi-head Attention Mechanism

Then, we divide the last dimensionality of $\tilde{Q}$, $\tilde{K}$, $\tilde{V}$ into two segments $\bar{Q}_{1,2}$, $\bar{K}_{1,2}$, $\bar{V}_{1,2} \in \mathbb{R}^{n \times \frac{m}{2}}$. We compute the dot products of the candidate token $\bar{Q}_{1,2}$ with other tokens $\bar{K}_{1,2}$, divide each by $\sqrt{d_k}$ (d_k is the dimensionality of $\tilde{K}$) and apply a softmax function to obtain the weights on $\bar{V}_{1,2}$:

$$\bar{O}_{1,2} = softmax(\frac{\bar{Q}_{1,2}\bar{K}_{1,2}^T}{\sqrt{d_k}})\bar{V}_{1,2} \tag{4}$$

Finally, we concatenate the outputs $\bar{O}_{1,2}$ into a new representation $O = [\bar{O}_1, \bar{O}_2]$ ($O \in \mathbb{R}^{n \times m}$), and concatenate O with the input representation X into the final output $T = [O, X]$ ($T \in \mathbb{R}^{n \times 2m}$). T is as the input of Bi-LSTM.

3) Bi-LSTM: Bi-LSTM [4] is a network that combines the forward Long Short-Term Memory (LSTM) [22] and the backward LSTM. It can capture bidirectional semantic dependencies, and combine context information effectively.

At every time step t, we set an input gate i_t, forget gate f_t, output gate o_t, and cell memory unit c_t. The LSTM unit obtains a distributed representation according to the current input T_t, the previous hidden layer state h_{t-1}, and the previous cell state c_{t-1}. The detailed operations of

LSTM are as follows:

$$\begin{bmatrix} i_t \\ f_t \\ o_t \\ \tilde{c}_t \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \sigma \\ tanh \end{bmatrix} \left(W \begin{bmatrix} T_t \\ h_{t-1} \end{bmatrix} + b \right) \quad (5)$$

$$c_t = i_t \odot \tilde{c}_t + f_t \odot c_{t-1}, h_t = o_t \odot tanh\left(c_t\right) \quad (6)$$

where $W \in \mathbb{R}^{2m \times d}$ is the weight matrix, $b \in \mathbb{R}^{n \times d}$ is the bias term, d is the size of the hidden units, σ refers to the sigmoid function, and $\odot$ denotes element-wise multiplication. Finally, the forward hidden state $\overrightarrow{h_t}$ and the backward hidden state $\overleftarrow{h_t}$ are concatenated into a single vector $h_t = [\overrightarrow{h_t}, \overleftarrow{h_t}]$ ($h_t \in \mathbb{R}^{2d}$). Then, we implement the multi-head attention on h_t and get a new sequence of vectors A. Finally, we concatenate H and A into a matrix $\tilde{H} = [H, A]$ ($\tilde{H} \in \mathbb{R}^{n \times 4d}$) as the input of a fully-connected layer.

4) Output: When identifying triggers, we use each token in the target sentence as the candidate trigger and calculate scores of event type for each token. Following Hong et al. [16], we use a fully-connected layer followed by a softmax function to obtain the predicted conditional probability $P(y|x, \theta)$ of different types:

$$P(y|x, \theta) = softmax(W\tilde{H} + b) \quad (7)$$

where $W \in \mathbb{R}^{4d \times t}$ is the weight matrix, $b \in \mathbb{R}^{n \times t}$ is the bias term, t represents the number of event types including one non-trigger type, and θ represents all parameters of the model.

5) Training: We minimize the objective function to reduce the cross-entropy loss of our model:

$$L(\Theta) = -\sum_{i=1}^{l}\sum_{j=1}^{n} \hat{y}_{ij} logP(y_j|x_i, \theta) \quad (8)$$

where $\hat{y}_{i_j}$ represents that the ith token x_i triggers the jth real event type, $P(y_j|x_i, \theta)$ represents scores of the jth predicted event type for the ith token x_i. We minimize the log-likelihood $L(\Theta)$ through Stochastic Gradient Descent (SGD) [10] optimizer to compute all the parameters θ.

V. Experiments

A. Experiment Settings

1) Dataset and Evaluation Metrics: Following the previous works [8], [11], [12], [23], we split ACE 2005 dataset into 40 documents for the test set, 30 documents for the development set, and 529 documents for the training set. Additionally, we utilize Precision (*P*), Recall (*R*), and F_1-score (F_1) as the evaluation metrics.

2) Hyper-parameters of Mention Segmentation: We adopt $BERT_{Large}$ (uncased)[4] for fine-tuning the pre-trained BERT model. The fixed length of sentences is set to 128, the batch size is set to 1, and the learning rate is set to 3e-5.

[4]https://storage.googleapis.com/bert_models/2018_10_18/uncased_L-24_H-1024_A-16.zip

3) Hyper-parameters of ED model: We use 300 dimensions for word embedding, 50 dimensions for entity type embedding, and 6 dimensions for event mention embedding. We set the fixed length of sentences as 80 by padding shorter sentences and cutting off longer sentences. We utilize LSTM with 100 hidden units. We also set the head number of multi-head attention as 2, and the hidden units of each-head attention as 100. The dropout rate is 0.2, the batch size is 10, and the learning rate is 0.3.

B. Overall Performance

We compare our performance with the following state-of-the-art methods:

- **Cross-Entity** [7] is a feature-based model based on cross-entity inference.
- **Cross-Event** [24] exploits relations among event types from the training set to help predict events in the test set.
- **DMCNN** [11] uses dynamic multi-pooling CNN to capture features.
- **Bi-RNN** [12] utilizes Bi-RNN to capture the dependencies of sequence.
- **PSL** [25] uses probabilistic soft logic to exploit latent and global information.
- **ANN** [20] uses argument information via supervised attention mechanisms.
- **GMLATT** [26] exploits cross-lingual attention to solve ambiguity problems.
- **HBTNGMA** [14] uses hierarchical and bias tagging networks with gated multi-level attention mechanisms.
- **HNN** [13] combines Bi-LSTM and CNN networks.

Table I shows the overall performance compared with the above state-of-the-art methods. These results demonstrate that our method is effective and the performance is better than all other state-of-the-art methods. F_1-scores of trigger identification and event type classification can reach 76.8% and 74.2% respectively, with a gain of 0.9% and 0.8% higher than the highest performance in other models. Furthermore, compared with other methods, the precision rate and recall rate of our proposed model is relatively balanced. The following reasons can explain why our method is better:

- Neural network-based models perform better than feature-based models because neural network-based models mitigate error propagation problems.
- In this task, Bi-LSTM can capture the context features and long-distance features of sentences more effective than Bi-RNN and CNN.
- Mention Segmentation can concentrate more on event-related information and reduce the interference of redundant words.
- Multi-head attention can get more attention to keywords in sentences and reduce the interference of meaningless words.

C. Effect of Event Mention

To confirm event mention help improve ED, we verify whether ground-truth event mention (ground-truth event

Table I
PERFORMANCE OF ALL ED SYSTEMS ON ACE 2005. (N/A: THE PAPER DIDN'T LIST RESULTS OF THIS TASK)

Methods	Trigger Identification			Type Classification		
	P	*R*	F_1	*P*	*R*	F_1
Cross-Entity [7]	n/a	n/a	n/a	72.9	64.3	68.3
Cross-Event [24]	n/a	n/a	n/a	68.7	68.9	68.8
DMCNN [11]	80.4	67.7	73.5	75.6	63.6	69.1
Bi-RNN [12]	68.5	75.7	71.9	66.0	73.0	69.3
PSL [25]	n/a	n/a	n/a	75.3	64.4	69.4
ANN [20]	n/a	n/a	n/a	76.8	67.5	71.9
GMLATT [26]	80.9	68.1	74.1	78.9	66.9	72.4
HBTNGMA [14]	n/a	n/a	n/a	77.9	69.1	73.3
HNN [13]	80.8	71.5	75.9	84.6	64.9	73.4
our model	77.2	76.5	**76.8**	74.1	74.3	**74.2**

Table II
PERFORMANCE OF GROUND-TRUTH EVENT MENTION AND PREDICTED EVENT MENTION IN DIFFERENT MODELS. * DENOTES THAT MENTION DERIVES FROM ACE 2005.

Methods	Identify Trigger			Classify Type		
	P	*R*	F_1	*P*	*R*	F_1
Bi-LSTM	72.0	77.4	74.6	68.6	73.8	71.1
+ground-truth mention*	87.2	86.2	**86.7**	82.8	81.8	**82.3**
Bi-LSTM+Att [27]	74.5	75.1	74.7	72.1	72.6	72.3
reproduced Bi-LSTM+Att	73.4	76.9	75.1	70.8	72.3	71.5
reproduced + mention	76.6	76.5	**76.5**	74.2	74.0	**74.1**
GAN [16]	75.3	78.8	77.0	71.3	74.7	73.0
reproduced GAN	75.2	76.5	75.8	72.1	73.3	72.7
reproduced + mention	79.2	75.0	**77.1**	76.7	72.6	**74.6**

mention is referred to annotated event mention in ACE 2005) is effective, and add predicted event mention to the reproduced previous works respectively.

1) Ground-truth Event Mention: Ground-truth event mention is concatenated into the input of Bi-LSTM. We set the Bi-LSTM as the baseline model, and add the ground-truth event mention as the feature to the Bi-LSTM. Experimental results in the top of Table II show that ground-truth event mention provides a large F_1 improvement of 12.1% and 11.2% over the baseline on ED task, and can fully indicate the effectiveness of ground-truth event mention.

2) Predicted Event Mention: Predicted event mention is added as a feature to the input of other models. We reproduce Bi-LSTM+Att [27] and GAN[16] model. Bi-LSTM+Att model uses entity relations, but we reproduce the model without using entity relations. The middle of Table II illustrates that predicted event mention on the Bi-LSTM+Att plays a critical role, yielding a 1.8% F_1 improvement for event type classification and trigger identification respectively. Furthermore, we conduct experiments on GAN model with predicted event mention, and the bottom of Table II shows that F_1-scores can reach 74.6% on type classification and 77.1% on trigger identification. Both can prove that the event mention is effective.

D. Effect of Multi-head Attention Mechanism

We utilize Bi-LSTM as the baseline model and add the multi-head attention mechanism to the Bi-LSTM. The result is shown in Table III, the F_1-scores of trigger identification and event type classification are both 0.5% higher than Bi-LSTM, indicating that the Bi-LSTM based on the multi-head attention is effective. Additionally, after incorporating predicted event mention and multi-head attention into the baseline, the F_1-scores of trigger identification and event type classification are 2.2% and 3.1% higher than the baseline, 1.1% and 1.3% higher than Bi-LSTM with predicted event mention. This indicates that multi-head attention is effective.

Table III
PERFORMANCE OF BI-LSTM, BI-LSTM WITH MULTI-HEAD ATTENTION, BI-LSTM WITH PREDICTED EVENT MENTION, AND MULTI-HEAD ATTENTION-BASED BI-LSTM WITH PREDICTED EVENT MENTION.

Methods	Trigger Identification			Type Classification		
	P	*R*	F_1	*P*	*R*	F_1
Bi-LSTM	72.0	77.4	74.6	68.6	73.8	71.1
+multi-head att	73.8	76.5	75.1	69.6	73.8	71.6
+mention	75.0	76.5	75.7	71.5	74.3	72.9
+multi-head att +mention	77.2	76.5	**76.8**	74.1	74.3	**74.2**

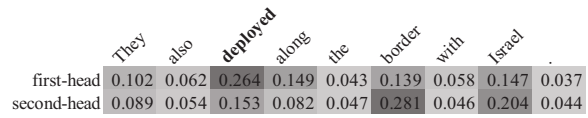

Figure 3. Visualization of the two-head attention results of "*deployed*"

We exploit two-head attention before and after the Bi-LSTM. Here, we will introduce how two-head attention works. As shown in sentence 3), the trigger word is "*deployed*", entities are "*They*", "*border*" and "*Israel*". Figure 3 shows the visualization of the two-head attention results of "*deployed*" in 3). As can be seen from Figure 3, the score of "*deployed*" in the first-head attention is the highest, indicating that the keyword "*deployed*" in the sentence is paid more attention by the attention mechanism. Furthermore, the scores of entities "*They*", "*border*" and "*Israel*" are higher than other words, indicating that the correlation between entities and triggers is relatively large and two-head attention can notice it. Thus, this also verifies the effectiveness of multi-head attention.

3) *They also* **deployed** *along the border with Israel.*

VI. CONCLUSION

In this paper, we propose a novel method for ED task, which extracts event mention and then uses mention segmentation to improve ED with multi-head attention-based Bi-LSTM. The purpose is to concentrate more on event-related information in sentences and avoid the interference of meaningless words. The experimental results demonstrate the effectiveness of our proposed method.

ACKNOWLEDGMENT

This work was supported by National Natural Science Foundation of China (Grants No. 61672368, 61672367, and 2017YFB1002104). The authors would like to thank the anonymous reviewers for their insightful comments

and suggestions. Yu Hong, Professor in Soochow University, is the corresponding author of the paper, whose email address is tianxianer@gmail.com.

REFERENCES

[1] S. Duan, R. He, and W. Zhao, "Exploiting document level information to improve event detection via recurrent neural networks," in *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, vol. 1, 2017, pp. 352–361.

[2] O. Vinyals, M. Fortunato, and N. Jaitly, "Pointer networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 2692–2700.

[3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[4] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[5] P. Gupta and H. Ji, "Predicting unknown time arguments based on cross-event propagation," in *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, 2009, pp. 369–372.

[6] R. Grishman, "The impact of task and corpus on event extraction systems." in *LREC*, 2010.

[7] Y. Hong, J. Zhang, B. Ma, J. Yao, G. Zhou, and Q. Zhu, "Using cross-entity inference to improve event extraction," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, 2011, pp. 1127–1136.

[8] Q. Li, H. Ji, and L. Huang, "Joint event extraction via structured prediction with global features," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, 2013, pp. 73–82.

[9] S. Liu, Y. Chen, S. He, K. Liu, and J. Zhao, "Leveraging framenet to improve automatic event detection," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, 2016, pp. 2134–2143.

[10] T. H. Nguyen and R. Grishman, "Event detection and domain adaptation with convolutional neural networks," in *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)*, vol. 2, 2015, pp. 365–371.

[11] Y. Chen, L. Xu, K. Liu, D. Zeng, and J. Zhao, "Event extraction via dynamic multi-pooling convolutional neural networks," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, vol. 1, 2015, pp. 167–176.

[12] T. H. Nguyen, K. Cho, and R. Grishman, "Joint event extraction via recurrent neural networks," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 300–309.

[13] X. Feng, B. Qin, and T. Liu, "A language-independent neural network for event detection," *Science China Information Sciences*, vol. 61, no. 9, p. 092106, 2018.

[14] Y. Chen, H. Yang, K. Liu, J. Zhao, and Y. Jia, "Collective event detection via a hierarchical and bias tagging networks with gated multi-level attention mechanisms," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 1267–1276.

[15] Y. Zhao, X. Jin, Y. Wang, and X. Cheng, "Document embedding enhanced event detection with hierarchical and supervised attention," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, vol. 2, 2018, pp. 414–419.

[16] Y. Hong, W. Zhou, G. Zhou, Q. Zhu *et al.*, "Self-regulation: Employing a generative adversarial network to improve event detection," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, 2018, pp. 515–526.

[17] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[18] H. Xu, B. Liu, L. Shu, and P. S. Yu, "Bert post-training for review reading comprehension and aspect-based sentiment analysis," *arXiv preprint arXiv:1904.02232*, 2019.

[19] G. R. Doddington, A. Mitchell, M. A. Przybocki, L. A. Ramshaw, S. M. Strassel, and R. M. Weischedel, "The automatic content extraction (ace) program-tasks, data, and evaluation." in *Lrec*, vol. 2, 2004, p. 1.

[20] S. Liu, Y. Chen, K. Liu, J. Zhao *et al.*, "Exploiting argument information to improve event detection via supervised attention mechanisms," 2017.

[21] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[22] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[23] S. Liu, R. Cheng, X. Yu, and X. Cheng, "Exploiting contextual information via dynamic memory network for event detection," *arXiv preprint arXiv:1810.03449*, 2018.

[24] S. Liao and R. Grishman, "Using document level cross-event inference to improve event extraction," in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2010, pp. 789–797.

[25] S. Liu, K. Liu, S. He, and J. Zhao, "A probabilistic soft logic based approach to exploiting latent and global information in event classification," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[26] J. Liu, Y. Chen, K. Liu, and J. Zhao, "Event detection via gated multilingual attention mechanism," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[27] J. Zhang, W. Zhou, Y. Hong, J. Yao, and M. Zhang, "Using entity relation to improve event detection via attention mechanism," in *CCF International Conference on Natural Language Processing and Chinese Computing*. Springer, 2018, pp. 171–183.