Document Embedding Enhanced Event Detection with Hierarchical and Supervised Attention

Yue Zhao, Xiaolong Jin, Yuanzhuo Wang, Xueqi Cheng
CAS Key Laboratory of Network Data Science and Technology,
Institute of Computing Technology, Chinese Academy of Science (CAS);
School of Computer and Control Engineering, The University of CAS
zhaoyue@software.ict.ac.cn;
{jinxiaolong, wangyuanzhuo, cxq}@ict.ac.cn

Abstract

Document-level information is very important for event detection even at sentence level. In this paper, we propose a novel Document Embedding Enhanced Bi-RNN model, called DEEB-RNN, to detect events in sentences. This model first learns event detection oriented embeddings of documents through a hierarchical and supervised attention based RNN, which pays word-level attention to event triggers and sentence-level attention to those sentences containing events. It then uses the learned document embedding to enhance another bidirectional RNN model to identify event triggers and their types in sentences. Through experiments on the ACE-2005 dataset, we demonstrate the effectiveness and merits of the proposed DEEB-RNN model via comparison with state-of-the-art methods.

1 Introduction

Event Detection (ED) is an important subtask of event extraction. It extracts event triggers from individual sentences and further identifies the type of the corresponding events. For instance, according to the ACE-2005 annotation guideline, in the sentence "Jane and John are married", an ED system should be able to identify the word "*married*" as a trigger of the event "*Marry*". However, it may be difficult to identify events from isolated sentences, because the same event trigger might represent different event types in different contexts.

Existing ED methods can mainly be categorized into two classes, namely, feature-based methods (e.g., (McClosky et al., 2011; Hong et al., 2011; Li et al., 2014)) and representation-based methods (e.g., (Nguyen and Grishman, 2015; Chen et al., 2015; Liu et al., 2016a; Chen et al., 2017)). The former mainly rely on a set of hand-designed features, while the latter employ distributed representation to capture meaningful semantic information. In general, most of these existing methods mainly exploit sentence-level contextual information. However, document-level information is also important for ED, because the sentences in the same document, although they may contain different types of events, are often correlated with respect to the theme of the document. For example, there are the following sentences in ACE-2005:

... I knew it was time to *leave*. Isn't that a great argument for term limits? ...

If we only examine the first sentence, it is hard to determine whether the trigger "*leave*" indicates a "*Transport*" event meaning that he wants to leave the current place, or an "*End-Position*" event indicating that he will stop working for his current organization. However, if we can capture the contextual information of this sentence, it is more confident for us to label "*leave*" as the trigger of an "*End-Position*" event. Upon such observation, there have been some feature-based studies (Ji and Grishman, 2008; Liao and Grishman, 2010; Huang and Riloff, 2012) that construct rules to capture document-level information for improving sentence-level ED. However, they suffer from two major limitations. First, the features used therein often need to be manually designed and may involve error propagation due to natural language processing; Second, they discover inter-event information at document level by constructing inference rules, which is time-consuming and is hard to make the rule set as complete as possible. Besides, a representation-based study has been presented in (Duan et al., 2017), which employs the PV-DM model to train document embeddings and further uses it in a RNN-based event classifier. However, as being limited by the unsupervised training

Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Short Papers), pages 414–419
Melbourne, Australia, July 15 - 20, 2018.

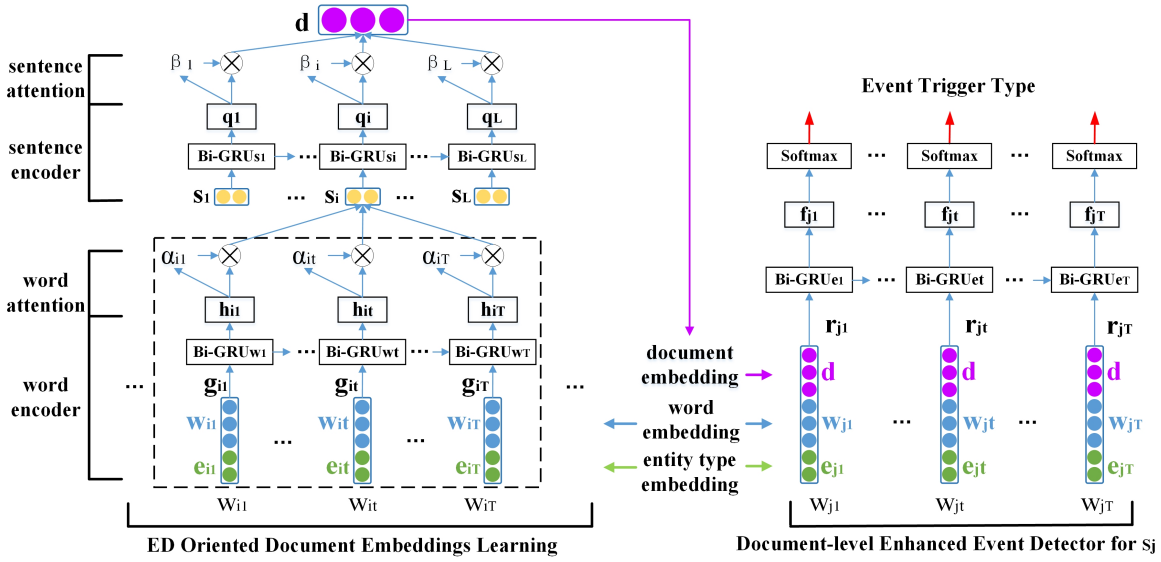

Figure 1: The schematic diagram of the DEEB-RNN model for ED at sentence level.

process, the document-level representation cannot specifically capture event-related information.

In this paper, we propose a novel Document Embedding Enhanced Bi-RNN model, called DEEB-RNN, for ED at sentence level. This model first learns ED oriented embeddings of documents through a hierarchical and supervised attention based bidirectional RNN, which pays word-level attention to event triggers and sentence-level attention to those sentences containing events. It then uses the learned document embeddings to facilitate another bidirectional RNN model to identify event triggers and their types in individual sentences. This learning process is guided by a general loss function where the loss corresponding to attention at both word and sentence levels and that of event type identification are integrated. It should be mentioned that although the attention mechanism has recently been applied effectively in various tasks, including machine translation (Zhang et al., 2017), question answering (Hao et al., 2017), document summarization (Tan et al., 2017), etc., this is the first study, to the best of our knowledge, which adopts a hierarchical and supervised attention mechanism to learn ED oriented embeddings of documents.

We evaluate the developed DEEB-RNN model on the benchmark dataset, ACE-2005, and systematically investigate the impacts of different supervised attention strategies on its performance. Experimental results show that the DEEB-RNN model outperforms both feature-based and representation-based state-of-the-art methods in terms of recall and F1-measure.

2 The Proposed Model

We formalize ED as a multi-class classification problem. Given a sentence, we treat every word in it as a trigger candidate, and classify each candidate to a certain event type. In the ACE-2005 dataset, there are 8 event types, further being divided into 33 subtypes, and a "Not Applicable (NA)" type. Without loss of generality, in this paper we regard the 33 subtypes as 33 event types. Figure 1 presents the schematic diagram of the proposed DEEB-RNN model, which contains two main modules:

1. The ED Oriented Document Embedding Learning (EDODEL) module, which learns the distributed representations of documents from both word and sentence levels via the well-designed hierarchical and supervised attention mechanism.

2. The Document-level Enhanced Event Detector (DEED) module, which tags each trigger candidate with an event type based on the learned embedding of documents.

2.1 The EDODEL Module

To learn the ED oriented embedding of a document, we apply the hierarchical and supervised attention network presented in Figure 1, which consists of a word-level Bi-GRU (Schuster and Paliwal, 2002) encoder with attention on event triggers

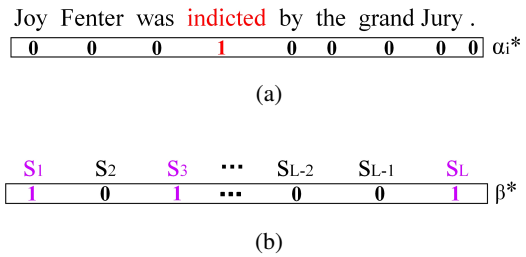

Figure 2: Examples of the gold word- and sentence-level attention without normalization. (a) Word-level attention. "*Indicated*" is a candidate trigger; (b) Sentence-level attention. The sentences in purple contain trigger words.

and a sentence-level Bi-GRU encoder with attention on sentences with events. Given a document with L sentences, DEEB-RNN learns its embedding for detecting events in all sentences.

Word-level embeddings Given a sentence s_i $(i = 1, 2, ..., L)$ consisting of words $\{w_{it}|t = 1, 2, ..., T\}$. For each word w_{it}, we first concatenate its embedding $\boldsymbol{w_{it}}$ and its entity type embedding[1] $\boldsymbol{e_{it}}$ (Nguyen and Grishman, 2015) as the input $\boldsymbol{g_{it}}$ of a Bi-GRU and thus obtain the bidirectional hidden state $\boldsymbol{h_{it}}$:

$$\boldsymbol{h_{it}} = [\overrightarrow{\text{GRU}_w}(\boldsymbol{g_{it}}), \overleftarrow{\text{GRU}_w}(\boldsymbol{g_{it}})]. \quad (1)$$

We then feed $\boldsymbol{h_{it}}$ to a perceptron with no bias to get $\boldsymbol{u_{it}} = \tanh(W_w \boldsymbol{h_{it}})$ as a hidden representation of $\boldsymbol{h_{it}}$ and also obtain an attention weight $\alpha_{it} = \boldsymbol{u_{it}}^{\mathrm{T}} c_w$, which should be normalized through a softmax function. Here, similar to that in (Yang et al., 2016), c_w is a vector representing the word-level context of w_{it}, which is initialized at random. Finally, the embedding of the sentence $\boldsymbol{s_i}$ can be obtained by summing up $\boldsymbol{h_{it}}$ with their weights:

$$\boldsymbol{s_i} = \sum_{t=1}^{T} \alpha_{it} \boldsymbol{h_{it}}. \quad (2)$$

To pay more attention to trigger words than other words, we construct the *gold word-level attention signals* $\boldsymbol{\alpha_i^*}$ for the sentence s_i, as illustrated in Figure 2a. We can then take the square error as the general loss of the attention at word level to supervise the learning process:

$$E_w(\boldsymbol{\alpha^*}, \boldsymbol{\alpha}) = \sum_{i=1}^{L} \sum_{t=1}^{T} (\alpha_{it}^* - \alpha_{it})^2. \quad (3)$$

[1]The words in the ACE-2005 dataset are annotated with their entity types (annotated as "NA" if they are not an entity).

Sentence-level embeddings Given the sentence embeddings $\{\boldsymbol{s_i}|i = 1, 2, ..., L\}$, we first get the hidden state $\boldsymbol{q_i}$ via a Bi-GRU:

$$\boldsymbol{q_i} = [\overrightarrow{\text{GRU}_s}(\boldsymbol{s_i}), \overleftarrow{\text{GRU}_s}(\boldsymbol{s_i})]. \quad (4)$$

Then we feed $\boldsymbol{q_i}$ to a perceptron with no bias to get the hidden representation $\boldsymbol{t_i} = \tanh(W_s \boldsymbol{q_i})$ and also obtain an attention weight $\beta_i = \boldsymbol{t_i}^{\mathrm{T}} c_s$ to be normalized via softmax. Similarly, c_s represents the sentence-level context of s_i to be randomly initialized. We eventually obtain the document embedding $\boldsymbol{d}$ as:

$$\boldsymbol{d} = \sum_{i=1}^{L} \beta_i \boldsymbol{s_i}. \quad (5)$$

We also think that the sentences containing event should obtain more attention than other ones. Therefore, similar to the case at word level, we construct the *gold sentence-level attention signals* $\boldsymbol{\beta^*}$ for the document d, as illustrated in Figure 2b, and further take the square error as the general loss of the attention at sentence level to supervise the learning process:

$$E_s(\boldsymbol{\beta^*}, \boldsymbol{\beta}) = \sum_{i=1}^{L} (\beta_i^* - \beta_i)^2. \quad (6)$$

2.2 The DEED Module

We employ another Bi-GRU encoder and a softmax output layer to model the ED task, which can handle event triggers with multiple words. Specifically, given a sentence s_j $(j = 1, 2, ..., L)$ in document d, for each of its word w_{jt} $(t = 1, 2, ..., T)$, we concatenate its word embedding $\boldsymbol{w_{jt}}$ and entity type embedding $\boldsymbol{e_{jt}}$ with the corresponding document embedding $\boldsymbol{d}$ as the input $\boldsymbol{r_{jt}}$ of the Bi-GRU and thus obtain the hidden state $\boldsymbol{f_{jt}}$:

$$\boldsymbol{f_{jt}} = [\overrightarrow{\text{GRU}_e}(\boldsymbol{r_{jt}}), \overleftarrow{\text{GRU}_e}(\boldsymbol{r_{jt}})]. \quad (7)$$

Finally, we get the probability vector o_{jt} with K dimensions through a softmax layer for w_{jt}, where the k-th element, $o_{jt}^{(k)}$, of o_{jt} indicates the probability of classifying w_{jt} to the k-th event type. The loss function, $J(\boldsymbol{y}, \boldsymbol{o})$, can thus be defined in terms of the cross-entropy error of the real event type y_{jt} and the predicted probability $o_{jt}^{(k)}$ as follows:

$$J(\boldsymbol{y}, \boldsymbol{o}) = -\sum_{j=1}^{L} \sum_{t=1}^{T} \sum_{k=1}^{K} \mathrm{I}(y_{jt} = k) \log o_{jt}^{(k)}, \quad (8)$$

where $I(\cdot)$ is the indicator function.

2.3 Joint Training of the DEEB-RNN model

In the DEEB-RNN model, the above two modules are jointly trained. For this purpose, we define the joint loss function in the training process upon the losses specified for different modules as follows:

$$\mathbb{J}(\theta) = \sum_{\forall d \in \phi} (J(\boldsymbol{y}, \boldsymbol{o}) + \lambda E_w(\boldsymbol{\alpha}^*, \boldsymbol{\alpha}) + \mu E_s(\boldsymbol{\beta}^*, \boldsymbol{\beta})), \tag{9}$$

where θ denotes, as a whole, the parameters used in DEEB-RNN, ϕ is the training document set, and λ and μ are hyper-parameters for striking a balance among $J(\boldsymbol{y}, \boldsymbol{o})$, $E_w(\boldsymbol{\alpha}^*, \boldsymbol{\alpha})$ and $E_s(\boldsymbol{\beta}^*, \boldsymbol{\beta})$.

3 Experiments

3.1 Datasets and Settings

We validate the proposed model through comparison with state-of-the-art methods on the ACE-2005 dataset. In the experiments, the validation set has 30 documents from different genres, the test set has 40 documents and the training set contains the remaining 529 documents. All the data preprocessing and evaluation criteria follow those in (Ghaeini et al., 2016).

Hyper-parameters are tuned on the validation set. We set the dimension of the hidden layers corresponding to GRU_w, GRU_s, and GRU_e to 300, 200, and 300, respectively, the output size of W_w and W_s to 600 and 400, respectively, the dimension of entity type embeddings to 50, the batch size to 25, the dropout rate to 0.5. In addition, we utilize the pre-trained word embeddings with 300 dimensions from (Mikolov et al., 2013) for initialization. For entity types, their embeddings are randomly initialized. We train the model using Stochastic Gradient Descent (SGD) over shuffled mini-batches and using dropout (Krizhevsky et al., 2012) for regularization.

3.2 Baseline Models

In order to validate the proposed DEEB-RNN model through experimental comparison, we choose the following typical models as the baselines.

Sentence-level is a feature-based model proposed in (Hong et al., 2011), which regards entity-type consistency as a key feature to predict event mentions.

Joint Local is a feature-based model developed in (Li et al., 2013), which incorporates such features that explicitly capture the dependency among multiple triggers and arguments.

Methods	λ	μ	P	R	F_1
Bi-GRU	-	-	66.2	72.3	69.1
DEEB-RNN	0	0	69.3	75.2	72.1
DEEB-RNN1	1	0	70.9	76.7	73.7
DEEB-RNN2	0	1	72.3	74.5	73.4
DEEB-RNN3	1	1	72.3	75.8	74.0

Table 1: Experimental results with different attention strategies.

JRNN is a representation-based model proposed in (Nguyen et al., 2016), which exploits the inter-dependency between event triggers and argument roles via discrete structures.

Skip-CNN is a representation-based model presented in (Nguyen and Grishman, 2016), which proposes a novel convolution to exploit non-consecutive k-grams for event detection.

ANN-S2 is a representation-based model developed in (Liu et al., 2017), which explicitly exploits argument information for event detection via supervised attention mechanisms.

Cross-event is a feature-based model proposed in (Liao and Grishman, 2010), which learns relations among event types from training corpus and futher helps predict the occurrence of events.

PSL is a feature-based model developed in (Liu et al., 2016b), which encods global information such as event-event association in the form of logic using the probabilistic soft logic model.

DLRNN is a representation-based model proposed in (Duan et al., 2017), which automatically extracts cross-sentence clues to improve sentence-level event detection.

3.3 Impacts of Different Attention Strategies

In this section, we conduct experiments on the ACE-2005 dataset to demonstrate the effectiveness of different attention strategies.

Bi-GRU is the basic ED model, which does not employ document-level embeddings.

DEEB-RNN uses the document embeddings and computes attentions without supervision, in which hyper-parameters λ and μ are set to 0.

DEEB-RNN1/2/3 means they uses the gold attention signals as supervision information. Specifically, DEEB-RNN1 uses only the gold word-level attention signal ($\lambda = 1$ and $\mu = 0$), DEEB-RNN2 uses only the gold sentence-level attention signal ($\lambda = 0$ and $\mu = 1$), whilst DEEB-RNN3 employs the gold attention signals at both word and sen-

Methods	P	R	F_1
Sentence-level (2011)	67.6	53.5	59.7
Joint Local (2013)	73.7	59.3	65.7
JRNN (2016)	66.0	73.0	69.3
Skip-CNN (2016)	N/A	N/A	71.3
ANN-S2 (2017)	78.0	66.3	71.7
Cross-event (2010)†	68.7	68.9	68.8
PSL (2016)†	75.3	64.4	69.4
DLRNN (2017)†	77.2	64.9	70.5
DEEB-RNN1†	70.9	76.7	73.7
DEEB-RNN2†	72.3	74.5	73.4
DEEB-RNN3†	72.3	75.8	74.0

Table 2: Comparison between different methods. † indicates that the corresponding ED method uses information at both sentence and document levels.

tence levels ($\lambda = 1$ and $\mu = 1$).

Table 1 compares these methods, where we can observe that the methods with document embeddings (i.e., the last four) significantly outperform the pure Bi-GRU method, which suggests that document-level information is very beneficial for ED. An interesting phenomenon is that, as compared to DEEB-RNN, DEEB-RNN2 changes the precision-recall balance. This is because of the following reasons. On one hand, as compared to DEEB-RNN, DEEB-RNN2 uses the gold sentence-level attention signal, indicating that it pays special attention to the sentences containing events with event triggers. In this way, the Bi-RNN model for learning document embeddings will filter out the sentences containing events but without explicit event triggers. That means the events detected by DEEB-RNN2 are basically the ones with explicit event triggers. Therefore, as compared to DEEB-RNN, the precision of DEEB-RNN2 is improved; On the other hand, the above strategy may result in less learning of words, which are event triggers but do not appear in the training dataset. Therefore, those sentences with such event triggers cannot be detected. The recall of DEEB-RNN2 is thus lowered, as compared to DEEB-RNN. Moreover, DEEB-RNN3 shows the best performance, indicating that the gold attention signals at both word and sentence levels are useful for ED.

3.4 Performance Comparison

Table 2 presents the overall performance of all methods on ACE-2005. We can see that different versions of DEEB-RNN consistently outperform the existing state-of-the-art methods in terms of both recall and F1-measure, while their precision is comparable to that of others. The better performance of DEEB-RNN can be explained by the following reasons: (1) Compared with feature-based methods, including *Sentence-level*, *Joint Local*, and representation-based methods, including *JRNN*, *Skip-CNN* and *ANN-S2*, our method exploits document-level information (i.e., the ED oriented document embeddings) from both word and sentence levels in a document by the supervised attention mechanism, which enhance the ability of identifying trigger words; (2) Compared with feature-based methods using document-level information, such as *Cross-event*, *PSL*, our method can automatically capture event types in documents via a end-to-end Bi-RNN based model without manually designed rules; (3) Compared with representation-based methods using document-level information, such as *DLRNN*, our method can learn event detection oriented embeddings of documents through the hierarchical and supervised attention based Bi-RNN network.

4 Conclusions and Future Work

In this study, we proposed a hierarchical and supervised attention based and document embedding enhanced Bi-RNN method, called DEEB-RNN, for event detection. We explored different strategies to construct gold word- and sentence-level attentions to focus on event information. Experiments on the ACE-2005 dataset demonstrate that DEEB-RNN achieves better performance as compared to the state-of-the-art methods in terms of both recall and F1-measure. In this paper, we can strike a balance between sentence and document embeddings by adjusting their dimensions. In the future, we may improve the DEEB-RNN model to automatically determine the weights of sentence and document embeddings.

Acknowledgments

This work is supported by National Key Research and Development Program of China under grants 2016YFB1000902 and 2017YFC0820404, and National Natural Science Foundation of China under grants 61772501, 61572473, 61572469, and 91646120. We are grateful to Dr. Liu Kang of the Institute of Automation, Chinese Academy of Sciences for very helpful discussion on event detection.

References

Yubo Chen, Shulin Liu, Xiang Zhang, Kang Liu, and Jun Zhao. 2017. Automatically labeled data generation for large scale event extraction. In *Association for Computational Linguistics*, pages 409–419.

Yubo Chen, Liheng Xu, Kang Liu, daojian zeng, and jun zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In *Association for Computational Linguistics*, pages 167–176.

Shaoyang Duan, Ruifang He, and Wenli Zhao. 2017. 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)*, pages 352–361.

Reza Ghaeini, Xiaoli Fern, Liang Huang, and Prasad Tadepalli. 2016. Event nugget detection with forward-backward recurrent neural networks. In *Association for Computational Linguistics*, pages 369–373.

Yanchao Hao, Yuanzhe Zhang, Kang Liu, Shizhu He, Zhanyi Liu, Hua Wu, and Jun Zhao. 2017. An end-to-end model for question answering over knowledge base with cross-attention combining global knowledge. In *Association for Computational Linguistics*, pages 221–231.

Yu Hong, Jianfeng Zhang, Bin Ma, Jianmin Yao, Guodong Zhou, and Qiaoming Zhu. 2011. Using cross-entity inference to improve event extraction. In *Association for Computational Linguistics*, pages 1127–1136.

Ruihong Huang and Ellen Riloff. 2012. Modeling textual cohesion for event extraction. In *AAAI*, pages 1664–1670.

Heng Ji and Ralph Grishman. 2008. Refining event extraction through cross-document inference. In *Association for Computational Linguistics*, pages 254–262.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, pages 1097–1105.

Qi Li, Heng Ji, Yu Hong, and Sujian Li. 2014. Constructing information networks using one single model. In *Empirical Methods in Natural Language Processing*, pages 1846–1851.

Qi Li, Heng Ji, and Liang Huang. 2013. Joint event extraction via structured prediction with global features. In *Association for Computational Linguistics*, pages 73–82.

Shasha Liao and Ralph Grishman. 2010. Using document level cross-event inference to improve event extraction. In *Association for Computational Linguistics*, pages 789–797.

Shulin Liu, Yubo Chen, Shizhu He, Kang Liu, and Jun Zhao. 2016a. Leveraging framenet to improve automatic event detection. In *Association for Computational Linguistics*, pages 2134–2143.

Shulin Liu, Yubo Chen, Kang Liu, and Jun Zhao. 2017. Exploiting argument information to improve event detection via supervised attention mechanisms. In *Association for Computational Linguistics*, pages 1789–1798.

Shulin Liu, Kang Liu, Shizhu He, and Jun Zhao. 2016b. A probabilistic soft logic based approach to exploiting latent and global information in event classification. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 2993–2999.

David McClosky, Mihai Surdeanu, and Christopher D. Manning. 2011. Event extraction as dependency parsing for bionlp 2011. In *Association for Computational Linguistics*, pages 41–45.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, pages 3111–3119.

Thien Nguyen, Kyunghyun Cho, and Ralph Grishman. 2016. Joint event extraction via recurrent neural networks. In *NAACL*, pages 300–309.

Thien Nguyen and Ralph Grishman. 2015. Event detection and domain adaptation with convolutional neural networks. In *IJCNLP*, pages 365–371.

Thien Nguyen and Ralph Grishman. 2016. Modeling skip-grams for event detection with convolutional neural networks. In *Empirical Methods in Natural Language Processing*, pages 886–891.

M. Schuster and K.K. Paliwal. 2002. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45:2673–2681.

Jiwei Tan, Xiaojun Wan, and Jianguo Xiao. 2017. Abstractive document summarization with a graph-based attentional neural model. In *Association for Computational Linguistics*, pages 1171–1181.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *NAACL*, pages 1480–1489.

Jinchao Zhang, Mingxuan Wang, Qun Liu, and Jie Zhou. 2017. Incorporating word reordering knowledge into attention-based neural machine translation. In *Association for Computational Linguistics*, pages 1524–1534.