Document Information Assisted Event Trigger Detection

1st Yingchi Liu
Alibaba Group
Bellevue, WA, USA
yingchi.liu@alibaba-inc.com

2nd Quanzhi Li
Alibaba Group
Bellevue, WA, USA
quanzhi.li@alibaba-inc.com

3rd Xiaozhong Liu
Alibaba Group
Bellevue, WA, USA
xiaozhong.lxz@alibaba-inc.com

4th Luo Si
Alibaba Group
Bellevue, WA, USA
luo.si@alibaba-inc.com

***Abstract*—Event trigger detection remains a challenging task. Most of previous studies focused on variations of model structures to extract features from the local context of the trigger words. However, few studies focused on the utilization of document level information. In this work, we studied the benefit of exploiting the document level information for event trigger detections in textual data. Two approaches of extracting document features are proposed, and the document features are integrated with the embeddings generated from the local context of the trigger word using a convolutional neural network (CNN) model. Our experiment shows that these two methods both outperform the CNN-based baseline model.**

***Index Terms*—event detection, event extraction, neural networks, cnn, rnn**

I. Introduction

Event extraction is a task to extract event triggers and arguments from textual data, which requires handling complex relations between entities. Event extraction researches aim to automatically extract such information from the text. It is essential in a variety of natural language processing applications including information retrieval, text summarization, knowledge base constructions and etc. For instance, it has been applied to identifying civilians killed by police [1] [2]. Event trigger detection is the key process for extracting events. As in the sentence, *"The Davao Medical Center, a regional government hospital, recorded 19 deaths with 50 wounded."* , the word "death" is the trigger word for the event type, "Life.die".

Technologies for automatic event extraction have been continuously advanced over the past years. Most of the earlier works [3] [4] [5] [6] considered event extraction as a classification problem and designed a lot of lexical and syntactic features. And in recent years, with its successes in many natural language processing tasks, neural network has been applied in event extractions as well. Convolutional neural network [7] was adapted for trigger detection [8], and recurrent neural network (RNN) [9] was used to joint learn the model for both event trigger detection and argument extraction [10].

Many variations of the neural networks has been studied. [11] [12] However, most of those models only focused on the use of limited local context around the target word. The document level information was under exploited. Duan et al. [13] proposed a document level recurrent neural network model for event detection under the assumptions: the word meaning can be affected by the context of the documents, the events from the same document are usually related to each other, such as Attack and Injury events, and documents sharing similar topics often have the same or related event types. In their model, the document level information was incorporated in document vectors pre-trained, which was later concatenated to word embedding of each word. In this paper, we propose two different approaches: 1) apply bi-directional LSTM [14] [15] to extract the document information during the training, 2) cosine similarities between documents are used to compute the event type scores for each document, and the resulting event type scores are considered as the document features and are attached to the local context features of the target word.

II. Models

Baseline Model : Owing to the capability of capturing local correlations of spatial or temporal structures, Convolutional neural networks (CNNs) [7] have achieved top performance in computer vision, speech recognition and natural language processings. For sentence modeling, CNNs perform excellently in extracting n-gram features at different positions of a sentence through convolutional filters, and can learn short and long-range relations through pooling operations. It has been demonstrated as an effective model for extracting semantic representations and capturing salient features in a flat structure [16]. Therefore, CNN has shown strong performances in many natural language tasks [17], especially in event trigger detection [8] [12]. In this work, we applied a simple structure of convolutional neural networks as our baseline (see Fig. 1).

As described by Nguyen et al. [8], the event trigger detection is formulated as a multi-class classification. A model is trained to predict the event type of each word in a document. For one word w, a fixed window of context around it, $(w_{-l}, w_{-l+1}, ...w_0, ...w_{l-1}, w_l)$, are used as input. The word sequence is transformed into real-value space by concatenation of word embeddings [18] and position embeddings [8]. The whole outputs of CNN are fed into the final layer of a feedforward neural network. The probability of each event type is calculated with the softmax function.

Bi-directional LSTM Extract Document Features : RNN [9] is another famous neural network architecture that has been successfully applied in many natural language processing tasks [19]. It is able to handle sequences of any length while

978-1-5386-5035-6/18/$31.00 ©2018 IEEE

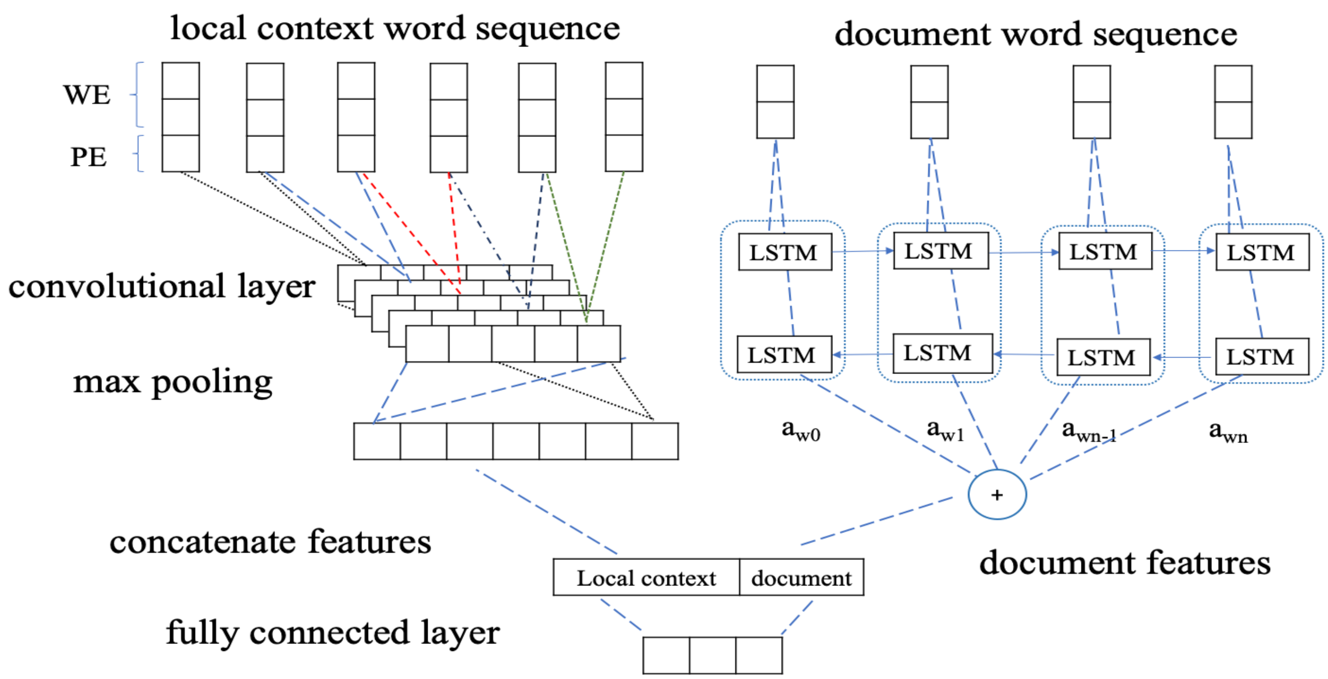

Fig. 1. Illustration of CNN with Document Features extracted by Bi-directional LSTM. "WE" stands for word embeddings and "PE" stands for position embeddings. a_{wi} represent the attention.

capturing long-term dependencies. To avoid the problem of gradient exploding or vanishing in the standard RNN, Long Short-term Memory (LSTM) [14] is frequently used. As its name suggested, bi-directional long short-term memory model [15] is a two-directional LSTM network with both forward and backward recurrent neural network, which can capture preceding and succeeding information. To extract document information and add it to the model, we used a bi-directional LSTM with attentions [20] to extract document features in this work. The documents were first converted as a sequence of word embeddings and passed into the bi-directional LSTM networks. The document features are represented by the attention weighted sum of the concatenated hidden states of the forward and backward layers, and hence appended to the outputs of the CNN. The whole concatenated feature vectors are passed into a final layer. The whole process is described in Fig. 1.

Document Event Scores as Features: There are others ways to represent a document. For example, we can also use topics generated from the document using LDA, or use document embeddings, such as paragraph vector or the average of all the word embeddings of this document. However, we want to explicitly relate event types with the documents and hence chose weighted event type scores as the document representations. Assuming there exists the consistency among the event types in the same document or related documents, we could assign scores for event types in a particular document. The scores of certain event types vary with the documents. To estimate those scores, we first computed the cosine similarities between the documents using the tf-idf weighted bag of word vector representation. As shown in (1), for a document *i*, the event type score S_{ik} of one even type *k* is calculated by summing over the occurrences of that event type *k* in all the training documents and weighed by the similarity between this document and each training document.

$$S_{ik} = \sum_{j} c_{ij} f_{jk} \tag{1}$$

The c_{ij} is the cosine similarity between document *i* and training document *j*, and

$$f_{jk} = \begin{cases} 1, & \text{if event k occurs in training document j} \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

Therefore, each document can be represented by a vector of event scores. Similarly, the document vectors are concatenated to CNN output and passed to the final output layer together.

III. Experiment

Data Set and Evaluation: In this work, we used ACE2005 event extraction data set. This data set includes 8 general event types and 33 event subtypes. Therefore, there are total 34 classes for the multi-class classification problem, along with event type "None". Follow previous work, we used the same 529 documents for training, 30 documents as development data and the rest 40 documents as test data [3] [8]. When evaluating the models, an event trigger is considered correct when the whole span of the trigger words are correctly predicted. For the evaluation metrics, we use F1 measure, which has been used in many previous studies [3] [8] [12] [13]

TABLE I
F SCORES FOR THE CNN MODEL, CNN MODEL + DOCUMENT FEATURES EXTRACTED BY BI-LSTM, CNN + DOCUMENT EVENT SCORES AS FEATURES

Models	Dev	Test
CNN	0.658	0.639
CNN + BiLSTM - Doc Feature	**0.680**	**0.654**
CNN + Doc Event Scores	0.665	0.649

Setup: A sequence words that are from context of the target word was used as input. The length of sequence was limited to 31 by using padding the shorter sentences or trimming off the longer ones. Word Embeddings in 300 dimensions are pre-trained and open sourced by Google.[1] As multiple window sizes in the convolutional layer has demonstrated benefits on sentence modeling [21]. In this experiment, we chose window sizes of {2, 3, 4, 5}, with a number of 150 filters for each. Position embedding size was chosen to be 10. Dropout rate was set as 0.5.

Results: As shown in Tab I, our baseline CNN received the event level f-score of 0.639. Bi-directional LSTM extracted document features have improved it to 0.654 and the features of document event scores increased the f score to 0.649. The document features did improve the performance of the model. Although the improvement with using event scores is less than that with the bi-directional LSTM neural networks, the process of getting event scores are simpler and faster.

IV. CONCLUSION

In this paper, we studied the event trigger detection problem. On top of a simple CNN structure, we provided document level information to assist the model to predict the event type for each word. Two approaches of extract document level information was presented. With those information, the performance of the event trigger detection model is improved, which demonstrated the potential of utilization of document level information to improve the performances event extraction or similar tasks. Effective approaches of extract document features can be further studied in the future research.

REFERENCES

[1] K. A. Keith, A. Handler, M. Pinkham, C. Magliozzi, J. McDuffie, and B. T. O'Connor, "Identifying civilians killed by police with distantly supervised entity-event extraction," in *EMNLP*, 2017.

[2] M. Nguyen and T. Nguyen, "Who is killed by police: Introducing supervised attention for hierarchical lstms," in *Proceedings of the 27th International Conference on Computational Linguistics*. Association for Computational Linguistics, 2018, pp. 2277–2287. [Online]. Available: http://aclweb.org/anthology/C18-1193

[3] H. Ji and R. Grishman, "Refining event extraction through cross-document inference," in *ACL-08: HLT - 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference*, 2008, pp. 254–262.

[4] 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*, ser. ACL '10. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010, pp. 789–797. [Online]. Available: http://dl.acm.org/citation.cfm?id=1858681.1858762

[1] https://github.com/mmihaltz/word2vec-GoogleNews-vectors

[5] 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*, ser. HLT '11. Stroudsburg, PA, USA: Association for Computational Linguistics, 2011, pp. 1127–1136. [Online]. Available: http://dl.acm.org/citation.cfm?id=2002472.2002615

[6] 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)*. Association for Computational Linguistics, 2013, pp. 73–82. [Online]. Available: http://aclweb.org/anthology/P13-1008

[7] Y. Lecun and Y. Bengio, *Convolutional networks for images, speech, and time-series*. MIT Press, 1995.

[8] T. H. Nguyen and R. Grishman, "Event detection and domain adaptation with convolutional neural networks," in *ACL*, 2015.

[9] T. Mikolov, M. Karafiát, L. Burget, J. Cernocký, and S. Khudanpur, "Recurrent neural network based language model," in *INTERSPEECH*, 2010.

[10] 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*. Association for Computational Linguistics, 2016, pp. 300–309. [Online]. Available: http://aclweb.org/anthology/N16-1034

[11] 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, Aug 2018. [Online]. Available: https://doi.org/10.1007/s11432-017-9359-x

[12] 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)*. Association for Computational Linguistics, 2015, pp. 167–176. [Online]. Available: http://aclweb.org/anthology/P15-1017

[13] 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)*. Asian Federation of Natural Language Processing, 2017, pp. 352–361. [Online]. Available: http://aclweb.org/anthology/I17-1036

[14] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997. [Online]. Available: https://doi.org/10.1162/neco.1997.9.8.1735

[15] M. Schuster, K. K. Paliwal, and A. General, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, 1997.

[16] Y. Liu, F. Wei, S. Li, H. Ji, M. Zhou, and H. Wang, "A dependency-based neural network for relation classification," *CoRR*, vol. abs/1507.04646, 2015. [Online]. Available: http://arxiv.org/abs/1507.04646

[17] Y. Kim, "Convolutional neural networks for sentence classification," *CoRR*, vol. abs/1408.5882, 2014. [Online]. Available: http://arxiv.org/abs/1408.5882

[18] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems 26*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2013, pp. 3111–3119. [Online]. Available: http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf

[19] W. Yin, K. Kann, M. Yu, and H. Schütze, "Comparative study of cnn and rnn for natural language processing," *CoRR*, vol. abs/1702.01923, 2017.

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *CoRR*, vol. abs/1706.03762, 2017. [Online]. Available: http://arxiv.org/abs/1706.03762

[21] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," *CoRR*, vol. abs/1404.2188, 2014. [Online]. Available: http://arxiv.org/abs/1404.2188