2021 International Conference on Culture-oriented Science & Technology (ICCST) | 978-1-6654-4254-1/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICCST53801.2021.00103

Movie Scene Argument Extraction with Trigger Action Information

1st Qian Yi
University of Chinese Academy of Sciences;
Institute of Automation, Chinese Academy of Science;
Beijing, China
yiqian2016@ia.ac.cn

2ndGuixuan Zhang
Beijing Engineering Research Center of Digital Content Technology,
Institute of Automation, Chinese Academy of Science;
Beijing, China

3rd Jie Liu
Beijing Engineering Research Center of Digital Content Technology,
Institute of Automation, Chinese Academy of Science;
Beijing, China

4th Shuwu Zhang
Beijing Engineering Research Center of Digital Content Technology,
Institute of Automation, Chinese Academy of Science;
Beijing, China

***Abstract*—Movie scene argument is an essential part of the movie scene. Movie scene argument extraction can help to understand the movie plot. In this paper we propose a movie scene argument extraction model, which utilizes the trigger action paraphrase as extra information to help improve the argument extraction. Specifically, we obtain the paraphrase of trigger from the dictionary and employ attention mechanism to encode them into an argument oriented embedding vector. Then we use the argument oriented embedding vector and the instance embedding for argument extraction. Experimental results on a movie scene event extraction dataset and a widely used open domain event extraction dataset prove effectiveness of our model.**

***Index Terms*—event extraction, attention, movie**

I. Introduction

Argument extraction is an important task in information extraction. It aims to extract the argument of the given event trigger and meanwhile identify the argument role. For example, in the sentence, *'Peter picks up the pistol and shoots Ruth."*, given the trigger *"shoots"* and the event type *"Exchange of Fire"*argument extraction need to extract the argument *"Peter"* and *"Rose"* and their argument role *"Attacker"* and *"Victim"*.

Since Event Extraction(EE) benefits many NLP applications [1–3], extensive efforts have been paid to event trigger extraction and event arguments extraction. Traditional feature-based methods[4–7] employed hand-crafted features and kernel-based classifier for classification. With the rapidly development of deep learning, neural networks methods have been adopted to automatically capture textual semantics features with low-dimensional vectors, and utilized these semantic features to extract event arguments, including convolutional neural networks(CNN) [8] and recurrent neural networks(RNN) [9, 10]. Advanced techniques also have been employed to further improve EE, such as zero-shot learning [11], multi-modal integration[12], and weakly supervised methods [13, 14].

Intuitive, the trigger word information can help to better understanding the instance's semantic context and the relationship between argument and triggers. So fusing trigger information into the instance representation can help to improve the performance of argument extraction.

In order to better exploit the trigger information, we first obtain the paraphrase of the given trigger word from Oxford advanced learner's dictionary. And for each candidate argument and trigger pair we adopt dynamic multi-pooling as the feature aggregator to obtain the instance representation. Then we calculate attention scores of the word in the paraphrase of the trigger with respect to the instance embedding vector and obtain the argument oriented trigger information embedding. Finally, we concatenate the argument oriented trigger information embedding and the instance embedding for argument extraction.

The contributions of this paper are as follow:

- We employ trigger word information to help to improve the argument extraction. And design a framework to effectively exploit the trigger word information.
- We construct a movie scene argument extraction dataset and verify the effectiveness of our model.

II. Methodology

In this section, we will introduce the structure of our model in detail, which consists of three main components: (1) The instance encoder first represents a sentence into hidden embeddings and use a sentence encoder to embed sentence semantic information into a low-dimensional continuous instance embedding. (2) The attention component builds an argument oriented embedding to model the trigger word information. (3) The argument role classifier using the concatenation of the instance embedding and the argument oriented trigger word vector as input to estimate the probability of a certain argument role for the instance.

A. Instance Encoder

An instance can be denoted as an n-word sequence $s = \{w_1, w_2, ..., t, ..., a, ..., w_{n-k}\}$, in which t, a represent the

978-1-6654-4254-1/21/$31.00 ©2021 IEEE
DOI 10.1109/ICCST53801.2021.00103

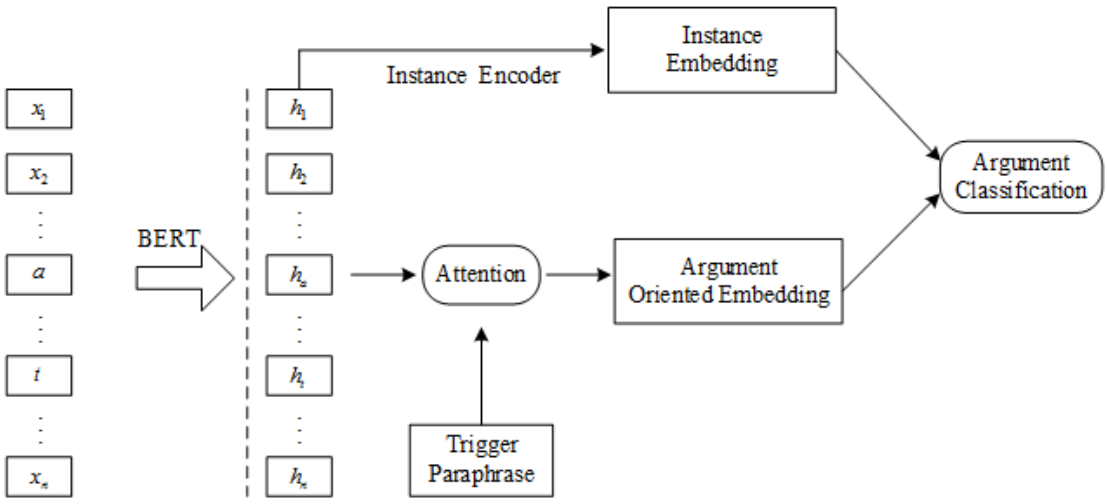

Fig. 1. The overall framework.

given event trigger and the candidate argument extracted by StanfordNER[16] respectively. The event trigger is extracted by the upstream event detection step (independent of our work) and each named entity detected by the NER tool in the sentence is a candidate argument.

1) Sentence Encoder: Sentence encoder aims to encode the sequence of words into their corresponding hidden embeddings:

$$\{h_1, h_2, ..., h_n\} = encoder(w_1, w_2, ..., t, ..., a, ..., w_n). \quad (1)$$

Here we use the BERT[15] as the encoder, which has achieve significant performance in other NLP tasks.

2) Instance encoder: As for the instance encoder, following the previous work[8] we use dynamic multi-pooling to aggregates the hidden embeddings matrix of input instance into an instance embedding:

$$\begin{aligned} {x^{(1)}}_j &= \max_{1 \leqslant i \leqslant i_t} \{h_{i,j},\} \\ {x^{(2)}}_j &= \max_{i_t < i \leqslant i_a} \{h_{i,j},\} \\ {x^{(3)}}_j &= \max_{i_a < i \leqslant n} \{h_{i,j},\} \\ x &= [x^{(1)}; x^{(2)}; x^{(3)}], \end{aligned} \quad (2)$$

where i is the index of word, i_t and i_a are the word index of trigger word and candidate argument word, and j is the feature index. We then concatenate the piecewise max-pooling results as the instance embedding x.

B. Trigger Paraphrase Embedding

In this section, we will introduce how to obtain the trigger paraphrase $p = \{q_1, q_2, ..., q_n\}$ into the argument oriented embedding. For each given trigger word t, we first find its paraphrase in the Oxford Learner's Dictionaries, which is an authoritative and widely used English dictionary. Specifically, we choose the first three paraphrase in each entry and from the paraphrase p for the trigger. Then we also utilize the BERT as the encoder, and embed the paraphrase sequence:

$$\{u_1, u_2, ..., u_n\} = \textit{BERT}(q_1, q_2, ..., q_n). \quad (3)$$

After obtaining the hidden embedding of trigger paraphrase, we calculate its attention score with respect to the given argument h_a:

$$\begin{aligned} a_i &= tanh(u_i^T \cdot W_a \cdot h_a); \\ s_{a_i} &= \frac{exp(a_i)}{\sum_{\mathcal{P}} exp(a_j)}, \end{aligned} \quad (4)$$

where W_a is a trainable weighted matrix and s_{a_i} is the attention score for the i-th word in trigger paraphrase. Finally, we obtain the argument oriented embedding by calculating the weighted sum of the hidden representations of paraphrase sequence:

$$e^{a_i} = \sum_{k=1}^{M} s_{a_i} \cdot u_k, \quad (5)$$

where M is the length of trigger paraphrase.

TABLE I
HYPER-PARAMETERS.

Parameter	Value
Learning Rate	0.001
Word Embedding Dimension	100
Convolutinal Output Dimension	300
Kernel Size	3
Batch size	25
Epoch size	20
Dropout rate	0.5
Learning rate	0.05
Optimizer	Adam

C. Argument Identification

In this section, to classify the argument role, we use the concatenation of instance embedding vector x and the

TABLE II
OVERALL RESULTS.

Model	ACE2005			MovieSceneEvent		
	Classification			Classification		
	P	R	F1	P	R	F1
JOINTFEATURE	64.7	44.4	52.7	48.9	30.1	34.6
DMCNN	62.2	46.9	53.5	50.1	36.7	42.4
dbRNN	66.2	52.8	58.7	50.6	44.3	45.7
Ours	66.9	51.1	**59.2**	52.3	45.6	**47.1**

argument-oriented trigger paraphrase embedding vector e^{a_i} as the input for the final argument role classifier. We feed the final instance representation into a feed forward layer with a sigmoid function. Finally, we adopt the Adam to minimize the negative log-likelihood loss to update the parameters.

III. EXPARIMENT AND DISCUSSION

A. Dataset

1) ACE2005: We adopt the most widely used dataset ACE2005 to evaluate our model, which contains 599 documents, which are annotated with 8 event types, 33 event subtypes, and 35 argument roles. Following the previous works [8], we use the same test set containing 40 documents, a development set with 30 randomly selected documents and training set with the remaining 529 documents.

2) MovieSceneEvent: We construct a movie scene event extraction dataset named MovieSceneEvent in this paper. It is constructed by manually labeling the movie scene sentences from the film scripts. It contains 5852 training samples and 486 testing samples with 12 event type and 18 argument roles.

B. Evaluation

We use the accuracy of argument classification as the metric to evaluate our model. An argument is correctly classified if its event subtype, span offsets and argument role match the annotation. And we present the result of argument classification experiment in the form of precision(P), recall(R), and F-measure(F1).

C. Experiment Setting

In this paper, the hyperparameter settings in experiment of our model are present in Table 1. As for the BERT for the sentence encoder, we use the BERT-BASE-CASED[15] model.

D. Result and Discussion

We compare our models with various baselines: (1) Feature-based methods, including Lis joint [7]. (2) DMCNN[8] proposed dynamic multi-pooling to aggregates the hidden embeddings. (3) Neural network with syntax information, like dbRNN[10] enhancing the recurrent neural network with dependency bridges to take syntactic information into consideration.

Experimental results are shown in Table 2. We can easily find out that our model do have some improvement and can steadily outperform the baseline models on both two datasets. It is worth to be noticed that the influence of trigger word information is more obvious on the movie scene dataset. We think it is because in movie scene situation the action information is more explicit, so the trigger word paraphrase is more influential.

When compared to DMCNN, the main difference between our model and DMCNN is that we introduce the trigger word information through attention. And our model can be seen as an extension of DMCNN. So the experimental results indicate that the trigger word information can benefit the argument classification.

IV. CONCLUSION

In this paper, we propose a movie scene argument extraction model, which introduce the trigger action paraphrase as extra information to help to improve the argument extraction. Specifically, we first obtain the paraphrase of given trigger word from a dictionary and adopt attention mechanism to fuse them into an argument oriented embedding vector. Then we utilize the concatenation of an argument-oriented embedding vector and the instance embedding vector for argument extraction.Experimental results on a movie scene event extraction dataset and widely used open domain dataset verify the effectiveness of our model.

ACKNOWLEDGEMENTS

This work was supported by the National Key R&D Program of China (2019YFB1406100) and Technology Program of Beijing (Z201100001820002). It was also the research achievement of the Key Laboratory of Digital Rights Services.

REFERENCES

[1] Yang, H. , Chua, T. S. , Wang, S. , Koh, C. K. . (2003). Structured use of external knowledge for event-based open domain question answering. ACM SIGIR FORUM(Special), p.33-40.

[2] Basile, P. , Caputo, A. , Semeraro, G. , Siciliani, L. . (2014). Extending an information retrieval system through time event extraction.

[3] Cheng, P. , Erk, K. . (2018). Implicit argument prediction with event knowledge.

[4] Patwardhan, Siddharth, Riloff, Ellen. (2009). A unified model of phrasal and sentential evidence for information extraction.

[5] Shasha Liao and Ralph Grishman. 2010b. Using document level cross-event inference to improve event extraction. In Proceedings of ACL, pages 789797.
[6] Huang, R. , Riloff, E. . (2013). In Proceedings of the 26th Conference on Artificial Intelligence (AAAI-12) Modeling Textual Cohesion for Event Extraction.
[7] Qi, L. , Ji, H. , Liang, H. . (2013). Joint Event Extraction via Structured Prediction with Global Features. Meeting of the Association for Computational Linguistics.
[8] Chen, Y. , Xu, L. , Kang, L. , Zeng, D. , Zhao, J. . (2015). Event Extraction via Dynamic Multi-Pooling Convolutional Neural Networks. The 53rd Annual Meeting of the Association for Computational Linguistics(ACL2015).
[9] Nguyen, T. H. , Cho, K. , Grishman, R. . (2016). Joint Event Extraction via Recurrent Neural Networks. Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.
[10] Sha, L. , Feng Qian, Baobao Chang, Sui, Z. . Jointly Extracting Event Triggers and Arguments by Dependency-Bridge RNN and Tensor-Based Argument Interaction.
[11] Huang, L. , Ji, H. , Cho, K. , Voss, C. R. . (2017). Zero-shot transfer learning for event extraction.
[12] Tongtao Zhang, Spencer Whitehead, Hanwang Zhang, Hongzhi Li, Joseph Ellis, Lifu Huang, Wei Liu, Heng Ji, and Shih-Fu Chang. 2017. Improving Event Extraction via Multimodal Integration. In Proceedings of the 25th ACM international conference on Multimedia (MM '17). Association for Computing Machinery, New York, NY, USA, 270278.
[13] Chen, Y. , Liu, S. , Xiang, Z. , Kang, L. , Zhao, J. . (2017). Automatically Labeled Data Generation for Large Scale Event Extraction. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers).
[14] Wang, X. , Han, X. , Liu, Z. , Sun, M. ,Li, P. . (2019). Adversarial Training for Weakly Supervised Event Detection. Proceedings of the 2019 Conference of the North.
[15] Devlin, J. , Chang, M. W. , Lee, K. , Toutanova, K. . (2018). Bert: pre-training of deep bidirectional transformers for language understanding.
[16] Jenny, R. F. , Trond, G. , Christopher, M. . (2005). Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005), 363-370. http://nlp.stanford.edu/ manning/papers/gibbscrf3.pdf