Proceedings of Machine Learning Research vol 245:1–7, 2024 2024 International Conference on Machine Learning and Intelligent Computing

Event Extraction in Complex Sentences Based on Dependency Parsing and Longformer

Lin Li LILIN1@CQUPT.EDU.CN
Chongqing University of Posts and Telecommunications

Ziyang Chen* S211201004@STU.CQUPT.EDU.CN
Chongqing University of Posts and Telecommunications

Shuxing Liao S221201022@STU.CQUPT.EDU.CN
Chongqing University of Posts and Telecommunications

Yibin Du S221231015@STU.CQUPT.EDU.CN
Chongqing University of Posts and Telecommunications

Hongxiao Wu S221231066@STU.CQUPT.EDU.CN
Chongqing University of Posts and Telecommunications

Zhihao Li S221231034@STU.CQUPT.EDU.CN
Chongqing University of Posts and Telecommunications

Editors: Nianyin Zeng and Ram Bilas Pachori

Abstract

Event extraction involves the identification and extraction of specific event-related information from a large corpus of textual data. In recent years, the introduction of pre-trained models has significantly enhanced the ability of these models to comprehend the semantics of sentences, leading to continuous advancements in event extraction methods. However, when it comes to long and complex sentences, these models have shown limited performance. This limitation can be attributed to the intricate structures of such sentences, which hinder the models' ability to grasp their semantic meaning. To tackle this challenge, we propose a novel model that combines dependency analysis tools with the Longformer pre-trained model. By effectively analyzing the structures of complex sentences, our model aims to enhance the semantic understanding of these sentences. Experimental results using the ACE2005 dataset demonstrate the improved performance of our model in event extraction for complex sentences.

Keywords: Event Extraction, Dependency Parsing, Pretrained Models

1. Introduction

All style elements are specified in this template to facilitate the production of your paper and to have the styles consistent throughout. The paragraph styles are built-in and examples of the styles are provided throughout this document. Save as you go and backup your work regularly! Event extraction is a fundamental task in various natural language processing applications, such as text summarization and reading comprehension. It involves automatically identifying and extracting events related to a specific topic or theme from a large corpus of textual data. In this study, our focus is specifically on event extraction tasks defined by the ACE (Automatic Content Extraction) conference. The current methods of event extraction still face the following challenges: Multiple expressions for the same event: A single event can be expressed using various trigger words, making it difficult to identify all possible triggers. 1) Ambiguity of trigger words: The same trigger word may have different meanings in different semantic contexts, requiring models to accurately interpret the intended meaning. 2) Complex sentence structures: The presence of complex sentence structures can hinder the model's semantic understanding capability, making it harder to capture the event-related information. 3) Addressing these challenges is essential to improve the performance of event extraction models and enhance their ability to accurately identify and classify events in text. In this paper, we propose a novel neural architecture called LADEE to overcome these limitations. We propose replacing the BERT model with the Longformer model (Beltagy et al., 2020) to reduce the capture of excessive redundant information in long sentences. We emphasize the importance of event trigger words and structural information from complex sentences. To address this, we use dependency parsing to analyze sentence structure and extract meaningful dependencies. We then utilize a self-attention mechanism inspired by the work proposed in (Yang et al., 2019) to effectively model the structural information using the pruned dependency parse graph. In summary, our work contributes the following:

- We propose a novel event extraction model, LADEE, based on dependency parsing and the Longformer model, to improve event extraction performance on long complex sentences.
- Experimental results on the ACE 2005 dataset demonstrate that LADEE achieves improved event extraction performance on long complex sentences.

2. Related Work

Dependency Parsing is a technique that shows how words in a sentence are connected, both syntactically and semantically. It uses a graph structure called a dependency parse tree, where the sentence is represented as a tree with words as nodes and relationships as directed edges. This method helps analyze the sentence's grammatical and semantic structure, offering a hierarchical view of its syntactic composition. Dependency trees indeed convey rich structural information and have shown to be valuable in various natural language processing tasks, including entity recognition. Previous studies, such as Nguyen et al. (2016), have demonstrated the effectiveness of utilizing dependency trees in entity recognition. Longformer Model is a variant of the Transformer model specifically designed to handle long texts. In contrast to the attention mechanism used in BERT models, Longformer introduces three novel attention patterns: sliding window attention, extended sliding window attention, and global + sliding window attention (Kim, 2014). These novel attention patterns in Longformer enable the model to effectively handle long texts by considering both local and global dependencies. The potential to understand long-distance relationships is especially crucial for tasks such as event extraction, where events may span across multiple sentences or paragraphs.

3. Methodology

As shown in Figure 1, our proposed LADEE framework for event extraction on a given sentence consists of two parts: (1) the left part - sequence encoder, and (2) the right part - structural encoding. In the following, we will provide detailed descriptions of each part.

3.1. Sequence Encoder

To address the issue of excessive redundant information in semantic features obtained by the BERT model, we propose the use of Longformer, a pretrained model specifically designed to handle long

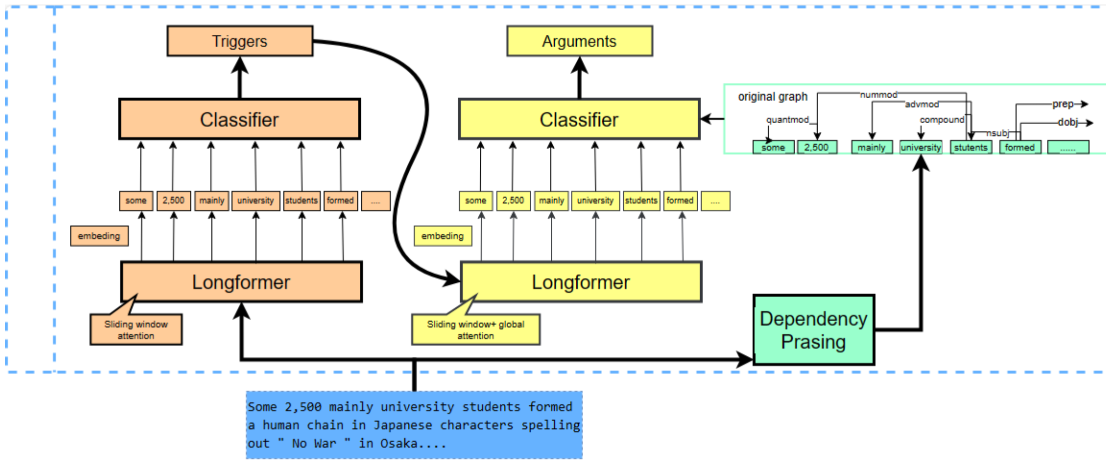

Figure 1: Model overall framework.

texts. The Longformer introduces a sliding window attention mechanism that reduces the inclusion of redundant information. By limiting the attention range within a fixed-size window, the Longformer mitigates the negative impact of excessive attention calculations and focuses on capturing the most relevant dependencies within the text. This approach helps improve the model's understanding of long and complex sentences. At the event detection stage, to encode a given sequence of words $X = (x_1, x_2, \ldots, x_n)$ using Longformer,.Obtain the token vector $T = (t_1, t_2, \ldots, t_n)$ for the event detection stage through the following formula (1).

$$(t_1, t_2, \ldots, t_n) = \text{longformer}(x_1, x_2, \ldots, x_n) \tag{1}$$

The input format for Longformer is similar to that of BERT, which includes word embeddings, segment embeddings, and position embeddings. Given that the input is a single sentence, all segment IDs are assigned a value of zero. Additionally, the beginning and ending positions of the sentence are marked with “ and ” tokens to indicate the start and end positions of the sentence. Frequently, the trigger word manifests as a phrase. Hence, we treat a sequence of adjacent tokens that share the same predicted label as a single, unified trigger word. In the event arguments extraction stage, we leverage the event type information obtained during the event detection stage. The detailed procedure encompasses merging annotated sentences with their corresponding event arguments, which are related to the event type and the sentences targeted for extraction. This combined input is then fed into the Longformer model to obtain semantic features anew. By utilizing the Longformer model's ‘global + sliding window’ attention mechanism, the sequence positions of the sentences to be extracted are treated with the global attention pattern, while the sequence positions of the annotated sentences with event arguments use the sliding window pattern, The calculation formula is as shown in (2):

$$(a_1, a_2, \ldots, a_n) = \text{longformer}(x_1, x_2, \ldots, x_n) \tag{2}$$

This approach ensures that the sentences $X = (x_1, x_2, \ldots, x_n)$ to be extracted can thoroughly explore their potential connections with the annotated sentences without dispersing the model's attention, Obtain the token vector representation $A = (a_1, a_2, \ldots, a_n)$ for the event arguments extraction stage.

3.2. Sentence Structure Encoder

Considering the specific nature of the event extraction task, which aims to identify key words or phrases indicating event occurrences in a sentence, we leverage the insights provided by the Dependency Parsing Graph(DPG). From the perspective of the DPG, we can understand that the target word necessarily has dependencies with other words in the sentence, signifying their roles and relationships. This aligns well with the need for a structured approach to sentence analysis, particularly in the more intricate event element extraction stage. For the modeling of structural information, we draw inspiration from the Transformer architecture (Vaswani et al., 2017) and design a similar attention-based modeling approach. Firstly, we generate three vectors, namely the key vector query vector, and value vector, based on the dependency relation labels obtained from the dependency parsing tool, The calculation formula is as shown in (3) $\sim$ (5):

$$\boldsymbol{k_i}^{\boldsymbol{l+1}} = W_k^l(\mathrm{node}_i^l) \tag{3}$$

$$\boldsymbol{q_i}^{\boldsymbol{l+1}} = W_q^l(\mathrm{node}_i^l) \tag{4}$$

$$\boldsymbol{node_i}^{\boldsymbol{i+1}} = W_v^l(\mathrm{node}_i^l) \tag{5}$$

In the formula, we utilize three vectors: the key vector W_k^l, the query vector W_k^l, and the value vector W_v^l, corresponding to the l-th layer. We initialize the dependency relation labels obtained from the dependency parsing tool with an embedding matrix node_i^0. Subsequently, the pruned dependency parsing graph is encoded using a neural network with L layers, resulting in feature representations $\mathrm{Node} = (\mathrm{node}_1^L, \mathrm{node}_2^L, \ldots, \mathrm{node}_n^L)$. These representations capture important information about the nodes and their dependencies. We then calculate weights to quantify the relationship between each node and its dependent nodes, The calculation formula is as shown in (6):

$$a_{ij} = \frac{\exp(\boldsymbol{q_i^{l+1}} \boldsymbol{k_j^{l+1}})}{\sum_{z \in N(i)} \exp(\boldsymbol{q_i^{l+1}} \boldsymbol{k_z^{l+1}})} \tag{6}$$

The attention score a_{ij} represents the attention score between node i and node j, where Z belongs to the set of all nodes that have a dependency relationship with node i.

$$node_i^{final} = tanh(node_i^{l+1} + \sum_{j \in N(i)} a_{ij} node_j^{l+1}) \tag{7}$$

Finally, as shown in formula (7), we obtain node i the final represents $node_i^{final}$ by integrating the fused features with the token embeddings through an additive operation and applying a hyperbolic tangent function tanh($\cdot$).

4. Experiments

4.1. Dataset and Evaluation Method

Our research involved carrying out experiments with the ACE 2005 corpus. The corpus underwent dependency relation analysis and pruning using the Stanza tool. Additionally, we adhere to the evaluation criteria established by previous research to assess the accuracy of the predicted event mentions (Xiang and Wang, 2019).

4.2. Experimental Results

We compared our framework with several previous competitive models for performance evaluation of ACE event extraction: Issa et al. (2018) represents a framework based on LSTM technology that utilizes information from dependency graphs for the extraction of event triggers and argument roles. DYGIE++ (Wadden et al., 2019) is a framework built upon BERT that focuses on modeling text spans while encompassing both intra-sentence and cross-sentence contextual information. OneIE (Lin et al., 2020) represents a neural-based joint model that incorporates global features for the purpose of extraction. EEQA (Du and Cardie, 2020) reinterprets the task of event extraction as a reading comprehension challenge by employing the BERT model. In Tables 1 and 2, we present a comparative analysis of the performance of these models in terms of trigger detection and argument extraction.

Table 1: Event detection results.

Method	Precision	Recall	F1
dbRNN	-	-	71.9
DYGIE++	-	-	68.9
OneIE	71.5	71.2	71.3
EEQA	69.5	75.9	72.6
Our LADEE	68.7	72.4	70.5

Table 2: Event arguments extraction results.

Method	Precision	Recall	F1
dbRNN	-	-	50.1
DYGIE++	-	-	52.5
OneIE	46.8	53.0	49.7
EEQA	56.8	50.2	53.3
Our LADEE	55.9	51.1	53.4

Regarding the effectiveness of each module, we conducted ablation experiments, and the results. are shown in Table 3. When using only the Longformer model to obtain semantic features, the precision (P) and recall (R) for the identification and classification of event arguments are 52% and 46.1%, respectively. After incorporating external annotated knowledge, precision and recall increased by 4.8% and 3.8%, and the F1 score improved by 4.2%. This indicates that the introduction of external annotated knowledge indeed helps the model better understand semantics. Finally, with the addition of structural feature information, although precision performance decreased by 0.9%, recall and F1 score increased by 1.2% and 0.3%, respectively. This suggests that structural feature information is also effective.

The analysis of the results from the ablation experiments reveals that as the sentence length increases and the sentence structure becomes more complex, existing methods experience a decrease in recall to varying degrees. In contrast, our model demonstrates better performance in addressing these challenges.

Table 3: Arguments extraction ablation experiments.

Index	Method	Precision	Recall	F1
1	Only Longformer	52.0	46.1	48.9
2	1+Annotated sentences	56.8	49.9	53.1
3	1+2+ Dependency	55.9	51.1	53.4

4.3. Limitations of the Method

Overall, the utilization of the Longformer model's 'global + sliding window' attention mechanism shows a noticeable improvement in event element extraction performance, particularly when incorporating annotated sentences for event arguments. However, the performance improvement is relatively small after adding structural information. This could be attributed to the modeling approach not effectively leveraging the structural information.

5. Conclusion and Outlook

This paper addresses the drawbacks of existing event extraction methods and proposes a novel redesign. The approach involves using a new pre-trained language model to overcome existing method limitations while preserving its advantages. Additionally, dependency parse trees are utilized to model sentence structure information. Experimental results on the ACE dataset confirm the proposed method's effectiveness. But there are still areas warranting further research. In future work, the primary focus will shift towards document-level event extraction. The benefits of the pre-trained language model Longformer can be more fully utilized in the analysis of longer documents. The absence of contextual information may lead to cases where sentence-level event extraction methods still struggle to identify event arguments. Incorporating contextual information aligns more closely with the human process of capturing events in text. This avenue will be a key focus in future research.

References

Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

Xinya Du and Claire Cardie. Event extraction by answering (almost) natural questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 671–683, 2020.

Fuad Issa, Marco Damonte, Shay B Cohen, Xiaohui Yan, and Yi Chang. Abstract meaning representation for paraphrase detection. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 442–452, 2018.

Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2014.

Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. A joint neural model for information extraction with global features. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 7999–8009, 2020.

Thien Huu Nguyen, Kyunghyun Cho, and Ralph 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*, pages 300–309, 2016.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30(2017), 2017.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. Entity, relation, and event extraction with contextualized span representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5784–5789, 2019.

Wei Xiang and Bang Wang. A survey of event extraction from text. *IEEE Access*, 7:173111–173137, 2019.

Sen Yang, Dawei Feng, Linbo Qiao, Zhigang Kan, and Dongsheng Li. Exploring pre-trained language models for event extraction and generation. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 5284–5294, 2019.