

Article

An Open-Domain Event Extraction Method Incorporating Semantic and Dependent Syntactic Information

Li He ^{1,2}, Qian Zhang ^{1,2,*}, Jianyong Duan ^{1,2} and Hao Wang ^{1,2}¹ College of Informatics, North China University of Technology, Beijing 100144, China² CNONIX National Standard Application and Promotion Laboratory, Beijing 100144, China

* Correspondence: zhangqianamy@163.com

Abstract: Open-domain event extraction is a fundamental task that aims to extract non-predefined types of events from news clusters. Some researchers have noticed that its performance can be enhanced by improving dependency relationships. Recently, graphical convolutional networks (GCNs) have been widely used to integrate dependency syntactic information into neural networks. However, they usually introduce noise and deteriorate the generalization. To tackle this issue, we propose using Bi-LSTM to obtain semantic representations of BERT intermediate layer features and infuse the dependent syntactic information. Compared to current methods, Bi-LSTM is more robust and has less dependency on word vectors and artificial features. Experiments on public datasets show that our approach is effective for open-domain event extraction tasks.

Keywords: semantic dependency syntax; graph convolution networks; open-domain event extraction



Citation: He, L.; Zhang, Q.; Duan, J.; Wang, H. An Open-Domain Event Extraction Method Incorporating Semantic and Dependent Syntactic Information. *Appl. Sci.* **2023**, *13*, 7942. <https://doi.org/10.3390/app13137942>

Academic Editor: Christos Bouras

Received: 20 May 2023

Revised: 30 June 2023

Accepted: 5 July 2023

Published: 6 July 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Open-domain event extraction tasks refer to extracting event information without pre-defined event types. Currently, the methods used in event extraction are based on semantic features [1], dependency syntactic structures [2], joint methods, knowledge graphs, etc. Although the semantic fusion method can capture semantic-rich features of the text sequence, the news cluster contains multiple events, so it is difficult for the model to fully and accurately identify all the events in the sentence. This is because the existing methods only use the sequence to represent the sentences, ignoring the syntactic structure, long-distance dependency, and global information of the text. It is difficult to capture long-distance information; in other words, the relevance of multiple events is lost.

Dependency syntax information has been widely used in natural language processing tasks. A large number of models and experiments have proven that the dependency relationship between words can effectively improve the performance of multiple tasks under various modeling methods, including event extraction tasks. The main method uses graphical convolutional networks (GCNs) to integrate dependency syntactic information into the deep neural network model, but this method has some noise, such as the influence of secondary dependency information, that is, no dependency relationship but with important features. A dependency attention-aware graphical convolutional network (DAGCN) [3], based on the dependency tree of dependency parsing, further uses the attention mechanism to increase the nodes' attention to key information and enhance the cohesion of the same parameter features, effectively using the main dependency relationships between nodes to learn syntactic structural information.

To address this challenge, we proposed using bidirectional long- and short-term memory (Bi-LSTM) to encode sentences to further enhance the semantic understanding of sentences, which have been proven to be suitable for capturing context-dependent semantic information. At the same time, we fused dependency syntactic information to obtain syntactic-rich structural information. Our work contributes in the following ways:

1. Semantic information is introduced based on BERT. We selected BERT end-layer features to obtain rich semantic features through Bi-LSTM and conducted experiments on the dataset to verify the validity of the semantic information;
2. Dependency syntactic information is further integrated. We analyzed the dependency syntactic information based on BERT middle layer features using the Stanford CoreNLP tool, further increasing the attention of nodes to the information in the graph through a DAGCN;
3. Semantic information and semantically enhanced dependency syntactic information are dynamically fused. We introduced a gating mechanism to reasonably control the information flow, extract rich and accurate feature representations, and again verify the feasibility of combining semantics and dependent syntax.

2. Related Work

In recent years, researchers have conducted several studies on open-domain event extraction tasks and achieved good results. The existing methods are mainly based on clustering [4], lexicon [5], Bayesian [6], and syntactic analysis. Clustering-based methods mainly construct event patterns for each event cluster by event pattern induction using slot values. Peng et al. [4] proposed a framework for media social event detection, a heterogeneous event-based information network and a novel GCN, based on which a parallel heterogeneous clustering algorithm (H-DBSCAN) was proposed for event detection and discovery.

Many other researchers have proposed lexicon-based approaches applied to open-domain event extraction tasks using word and phrase vocabularies to support sequential event extraction tasks. For example, Arnulphy et al. [7] used patterns and shallow parsing to automatically construct a noun event extraction lexicon. Vroe et al. [5] proposed MONTEE, which is able to distinguish between different types of patterns. That is, to determine whether an event has occurred, has not occurred, or is uncertain, helping to avoid extracting untrue events. Most Bayesian-based approaches assume that documents are a joint distribution of different types of events, slots, entities, and contextual features. For example, Wang et al. [6] proposed an open-event extraction model based on Bayesian and generative adversarial networks. Specifically, Dirichlet prior distributions and generators were used to identify potential event patterns. Zhou et al. [8] proposed a Bayesian model for extracting structured event representations from media, the latent event model (LEM), which is an unsupervised approach that does not require annotated data. Xiao Liu et al. [9] proposed the deep latent variable model (DLVM), which is a neural topic model that was used to extract event types, patterns, and parameters with better performance than other basic models. However, most of these approaches ignore semantic and syntactic structural information.

The results of syntactic analysis are often used to improve open-domain event extraction methods. For example, verbs help in event trigger detection, while nouns help in event parameter filtering. Syntactic dependencies, on the other hand, are useful for obtaining roles and parameters for the same event in multiple sentences. For example, Chau et al. [10] used syntactic analysis, WordNet, and word disambiguation tools on a gasoline price prediction task to extract relevant events from the news to be input into a deep neural network. Research has shown that the utilization of dependent syntactic information in open-domain event extraction is significantly reduced due to the contextual features obtained by pre-trained encoding already containing certain structural information. However, approaches incorporating dependent syntactic information have been shown to be effective in improving the performance of tasks such as named entity recognition, biomedical event extraction tasks, and sentiment classification. For example, Bohnet et al. [11] redefined named entity recognition as a structured prediction task based on the idea of dependency resolution in graphs. Kilicoglu et al. [12] used dependency resolution to analyze dependency paths on a biomedical event extraction task to discover regularities in biological texts using syntactic patterns. Therefore, in order to obtain the best extraction results, not only semantic but

also dependency syntactic information is needed. In this paper, we encode contextual features based on pre-trained models, learn semantic representations through the Bi-LSTM, use intermediate layer features to obtain syntactic information using dependent syntactic tools, and further realize modeling through DAGCNs with the gating mechanism used to dynamically fuse semantic information and semantically enhanced dependent syntactic information to obtain richer and more accurate features to improve the performance of open-domain event extraction.

3. Our Approach

This section studies the open-domain event extraction method, which combines semantics and dependency syntax. As mentioned earlier, the current event extraction methods tend to discard semantic information, resulting in poor extraction accuracy. Therefore, we propose dynamically merging semantic and dependency syntactic information from two perspectives. The overall architecture is shown in Figure 1.

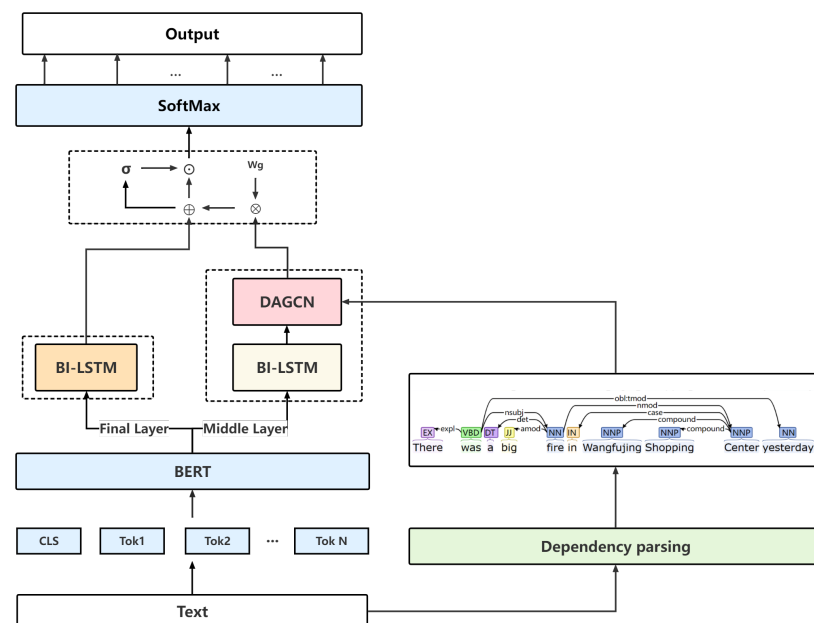


Figure 1. Architecture diagram of the open-domain event extraction model integrating semantic and dependency syntax information.

In this work, the feature representation of the last layer is selected on the basis of BERT, and rich semantic features are obtained by using Bi-LSTM.

Dependency syntactic features are further integrated to solve the problem of difficult-to-capture remote information. At the same time, DAGCN is used to model the graphical information, realize the direct use of dependency syntactic information, and focus on more critical node information, thus reducing the noise of dependency information. Specifically, in order to avoid the interaction between the syntactic structural information contained in the final word embedding and the dependency syntactic information, BERT non-adjacent middle layer features are selected through Bi-LSTM as the input of the DAGCN to obtain richer semantically enhanced dependency syntactic information.

Inspired by previous work, this paper uses a gating mechanism to dynamically fuse the semantically enhanced representation based on the final layer and the semantically feature-enhanced dependent syntactic representation based on the intermediate layer to obtain a richer and more accurate feature representation, further improving the performance of open-domain event extraction and laying a solid foundation for event mapping construction.

3.1. Embedding Layer

At the embedding layer, we input statements into the pre-trained language model BERT for coding, and obtain the context features for each layer. As shown in Equation (1), we obtain the sentence representation according to the input processing of BERT as follows:

$$W_{\text{BERT}} = \{[\text{CLS}], w_1, w_1, \dots, w_i, \dots, w_n, [\text{SEP}]\} \quad (1)$$

where n is the sentence length, $[\text{CLS}]$ is the first symbol, and $[\text{SEP}]$ marks the end of the statement. A sequence of length m is generated after the word segmentation $X = x_1, x_2 \dots x_m$ ($n \leq m$).

Secondly, considering that the input of the BERT model has three kinds (word vector, position vector, and piecewise vector), word vector refers to the embedding vector of each word, position vector represents the position of each word in the input sequence, and piecewise vector is used to distinguish different sentences. These three embedding vectors can be added because their shapes are the same (sequence length, embedding dimension), so we add the word vector, the position vector, and the segment vector as the input of the BERT coding layer to better represent the words in the input sequence. Finally, the coding information of the BERT layer can be obtained through the context feature transformed by the embedding layer coding as H_x^i . The equation is as follows:

$$h_{\text{token } j}^i = \text{BERT}^i(\text{token } j) \quad (2)$$

$$H_x^i = (h_{x_1}^i, h_{x_2}^i, \dots, h_{x_m}^i) \quad (3)$$

3.2. Semantic Enhancement Presentation Layer Based on the Final Layer Features

As shown in Figure 2, the BERT model consists of a 13-layer transformer encoder structure. However, multi-layer transformer coding can cause gradient loss or explosion problems during training, causing further loss of semantic information. The LSTM model has been proven to be an effective method to solve this problem. However, due to the problem that LSTM cannot capture reverse sequence information, researchers have proposed combining forward and backward LSTM (as shown in Figure 3) to obtain bidirectional text representation and have achieved good results in various NLP tasks.

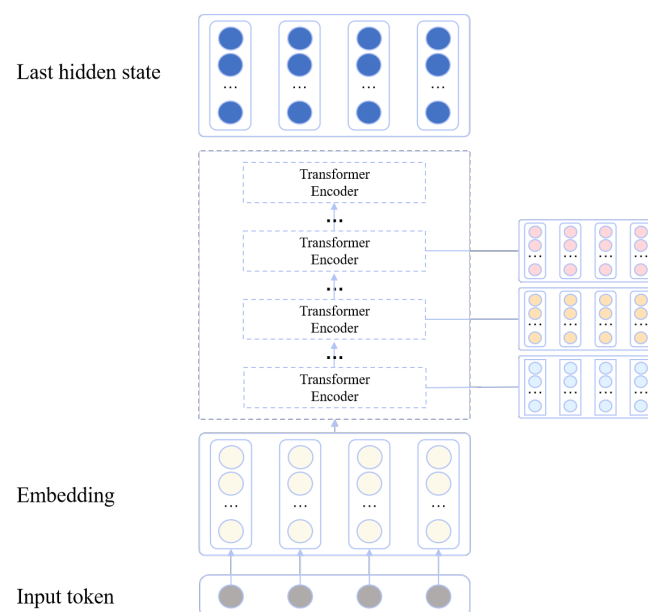


Figure 2. Schematic of the BERT model.

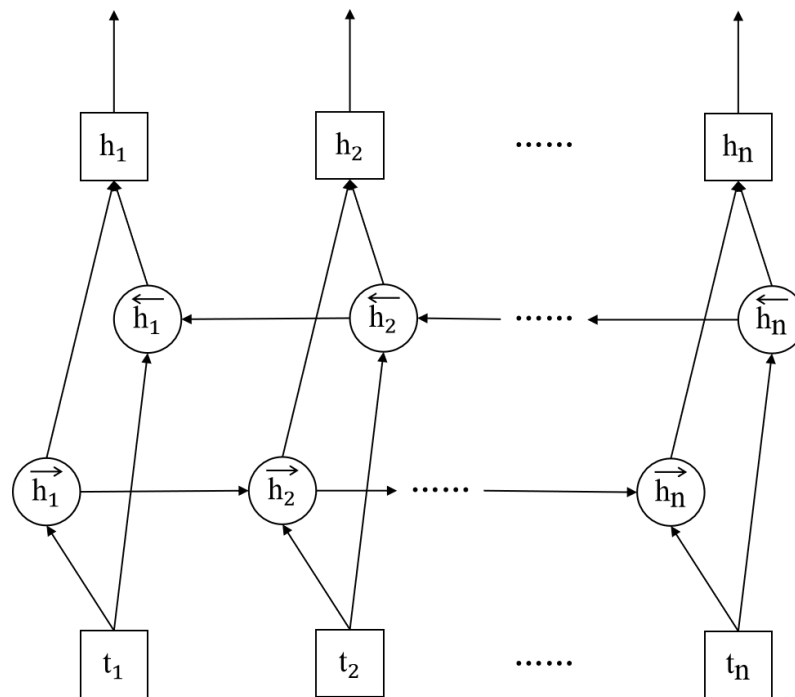


Figure 3. Bidirectional LSTM diagram.

In order to obtain the deep semantic representation and context dependence of sentences, this excerpt uses the Bi-LSTM network to learn the sentence's semantic representation. The feature representation of the last layer is selected on the basis of BERT, and the sequence information of sentences is captured by Bi-LSTM to obtain a rich semantic feature representation M_1 , as shown in Equation (4):

$$M_1 = \text{BiLSTM}(H_x^{13}) \quad (4)$$

where H_x^{13} is the output result of the BERT final layer, and M_1 includes the concatenation result of the forward and backward LSTMs.

3.3. Semantic Enhancement Dependency Syntax Representation Layer Based on Middle Layer Features

After obtaining the semantic information of the final feature, this section obtains the dependent syntactic features of semantic enhancement from another dimension. Firstly, to avoid the mutual misdirection of some syntactic structural information contained in the final layer of BERT and the dependency syntax information, the hidden state of the BERT non-adjacent intermediate layer is obtained as the input of the Bi-LSTM to implement coding, providing word representation to model the dependency syntax and avoid capturing information similarity.

Previous studies have shown that the selection of the middle layer can affect model performance. Considering the difference in the performances of different downstream tasks set in the middle layer, this paper conducted experiments on open-domain event extraction tasks. After experimental verification, layers 1, 4, 7, and 10 showed the best performance. Therefore, based on the first section, features $[H_x^1, H_x^4, H_x^7, H_x^{10}]$ were input into the Bi-LSTM to enhance information representation. In the input process, since BERT encodes tokens not as the original words but as word pieces, the first token embedding is used as the word embedding, and we assume that word x is divided into four subwords x_1, x_2, x_3 , and x_4 ; then, x_1 is the first token embedding. Equation (5) is the embedding vector representation of the word x . Equation (6) shows the specific approach: the location of the head word is recorded using the $head_{indices}$ matrix; subsequently, the corresponding location is embedded based on the token sequence.

$$p_x = h_{x_1} \quad (5)$$

$$P_w = [h_{w1}, \dots, h_{wn}] = \text{head}_{\text{indices}(h_{x1}, \dots, h_{xm})} \quad (6)$$

The coding process of the Bi-LSTM is shown as follows, where Equation (7) is the result of the forward LSTM, and Equation (8) is the result of the backward LSTM. Together, the results are concatenated to obtain $M_2 = [\vec{H}_{\text{Bi-LSTM}} \parallel \overleftarrow{H}_{\text{Bi-LSTM}}]$.

$$\vec{H}_{\text{Bi-LSTM}} = [\vec{p}_{w1}, \dots, \vec{p}_{wn}] = \overrightarrow{\text{LSTM}}(p_{w1}, \dots, p_{wn}) \quad (7)$$

$$\overleftarrow{H}_{\text{Bi-LSTM}} = [\overleftarrow{p}_{w1}, \dots, \overleftarrow{p}_w] = \overleftarrow{\text{LSTM}}(p_{w1}, \dots, p_{wn}) \quad (8)$$

After obtaining the semantic enhancement information based on the middle layer feature, this method also introduces syntactic structural information. The feature representation is input into the DAGCN to realize the direct use of dependency syntactic information and focus more on critical node information to avoid the noise problem. Finally, the dependency syntactic information representation based on the feature semantic enhancement of the middle layer is obtained. Specifically, a matrix $\hat{A}$ is first defined, corresponding to the dependency label for the dependency graph (A refers to the graph adjacency matrix). Then, all dependency tags are trained to generate the corresponding transition matrix TAG on the basis of this matrix, and $\hat{A}$ is transformed to obtain $\tilde{A}$. Finally, a weighted transformation is performed on the adjacent node values of elements in the input matrix H^l to update the node values, and the output of the GCN at layer $l+1$ can be obtained through the activation function. The specific calculation equation is as follows:

$$H^{l+1} = \sigma\left(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^l W^l\right) \quad (9)$$

where $\tilde{D}$ is a diagonal matrix, H^l is the output feature of the GCN of the previous layer, W^l is the parameter matrix of the GCN, where $\hat{A} \in R^{n \times n}$, $\tilde{D} \in R^{n \times n}$, $H^l \in R^{n \times d}$, $W^l \in R^{d \times d}$, n represents the number of nodes in the figure, and d represents the dimension of the GCN vector.

Note that this method uses dependency syntactic trees for graphical convolution. The complete dependency syntax tree was obtained by the Stanford CoreNLP dependency syntax tool [13]. As shown in Figure 4, each word is a node, and the edge represents the dependency relationship between the words. The adjacency matrix contains the dependency relationships of the dependency tree, and the syntactic information is further transformed into directed graph information through the adjacency matrix.

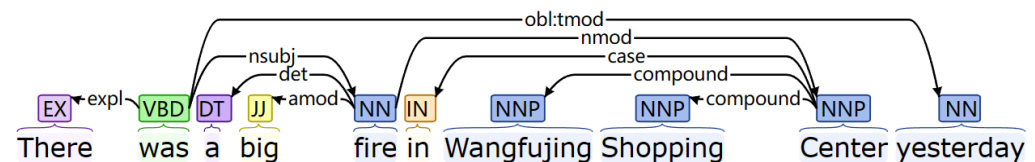


Figure 4. Dependent syntax diagram.

Stanford CoreNLP has more than 50 dependencies, and Table 1 illustrates some of the major dependencies with dependency tags, descriptions, and some examples given for illustration.

Table 1. Description of the dependency relationship.

Tag	Description	Explanation	Case
expl	expletive	The main verb of the clause	“There is a ghost in the room” expl (is, There)
tmod	temporal modifier	Time modification	“Last night, I swam in the pool” tmod (swam, night)
nsubj	nominal subject	Nominal subject	“Clinton defeated Dole” nsubj (defeated, Clinton)
det	determiner	Determiner	“The man is here” det (man, the)
amod	adjectival modifier	A descriptive modifier that modifies a noun phrase	“Sam eats red meat” amod (meat, red)
acomp	adjectival complement	A form complement used in verbs	“She looks very beautiful”. acomp (looks, beautiful)
advcl	adverbial clause modifier	An adverbial clause that modifies a verb	“The accident happened was falling” advcl (happened, falling)
doobj	direct object	Direct object	“She gave me a raise” dobi (gave, raise)
nsubjpass	passive nominal subject	Passive noun subject	“Dole was defeated by Clinton” nsubjpass (defeated, Dole)

3.4. Semantic Representation and Enhancement Depend on the Syntactic Representation Fusion Layer

Semantic features are part of the syntactic structure, and syntactic structural information is helpful in semantic learning, so it is necessary to integrate semantic information and dependency syntactic information. A gate mechanism can not only reduce the impact of noise by controlling the flow of information with a large data volume but can also learn different features to improve the robustness of the model.

Two aspects of feature representation have been obtained from the second and third sections, namely semantic information representation based on the final layer feature and semantically enhanced dependency syntactic information representation based on the middle layer feature. Inspired by previous work, this paper uses a gating mechanism to further dynamically integrate the two feature representations to achieve two-way weighting, reasonably allocate semantic and dependency syntax representations, and extract rich and accurate feature representations. This is shown in Equations (10) and (11) ($\oplus$ represents concatenation, W_g represents the weight matrix, $\odot$ represents element multiplication, and σ represents the activation function sigmoid):

$$K = [W_g \times H^{l+1} \oplus M_1] \quad (10)$$

$$O = \sigma(K) \odot K \quad (11)$$

4. Experiments

4.1. Dataset

This study used the GNBUSINESS dataset, which contains business news stories from 17 October 2018 to 22 January 2019, retaining the headlines and first paragraphs of the news stories in each news cluster, totaling 55,618 news stories, 13,047 news clusters, and 288 batches. In each news cluster, there are no more than 5 news reports, and 18 batches of news clusters were randomly selected, totaling 680 clusters, which were divided into the training set and test set at a 1:5 ratio.

4.2. Experimental Setup

In this experiment, the deep learning framework Pytorch was used to build, train, and test the model framework on a GPU. The experimental parameters are as follows:

The hidden element vector dimension of the Bi-LSTM was 250, the graphical convolution hidden element dependent on attention perception was 150, the number of layers was 2, the maximum epoch was 80, the batch size was 60, the dropout rate was 0.19, and the learning rate was 5×10^{-4} .

4.3. Experimental Results and Analysis

4.3.1. Comparison Experiments

In order to verify the validity of the proposed model, the following models were selected for experiments on the dataset as the main experiment objects, which are mainly divided into three categories: the model based on semantic analysis, the model based on dependency parsing, and the model based on joint semantic syntax.

Firstly, the Bi-LSTM model was selected based on semantic analysis. This model combines forward and backward LSTM, which can consider contextual information when processing sequence data and capture long-distance dependency characteristics.

Secondly, DBRNN [14], GCN, GCN-ED [2], and DAGCN [3] were used to model dependency parsing. Specifically, based on Bi-LSTM, DBRNN integrates dependency syntactic information and adds a dependency bridge for event extraction tasks. However, the GCN modeling concatenates the word vector and entity-type vector as the input to the GCN to improve the event extraction performance. Since the GCN only considers the relationship between nodes, researchers proposed a variant of the GCN, the GCN-ED, based on this model. In addition to the embedding vector of nodes, the adjacency matrix was also used to calculate the embedding vector of the edge. Because it considers the relationship between nodes and edges, it can better capture complex relationships. In addition, the DAGCN enhances the differentiation of different features to obtain key information features.

Finally, a model combining multiple neural network structures proposed by [15,16] was selected. Yuanfang Yu et al. [16] selected BERT intermediate layer features to reduce the interaction between the semantic and dependency syntax and improve the extraction accuracy, while GFDS [15] proposed a method to dynamically integrate semantic and dependency syntax information using a gating mechanism. The experimental results are shown in Table 2.

Table 2. Experiment results.

Method	Scheme Matching (%)		
	<i>P</i>	<i>R</i>	<i>F₁</i>
Bi-LSTM	60.5	41.6	49.3
DBRNN [14]	52.1	47.7	49.8
GCN-ED [2]	53.1	49.8	51.4
DAGCN [3]	53.5	50.4	51.9
Yuanfang Yu et al. [16]	52.6	49.7	51.1
GFDS [15]	52.5	51.9	52.2
Our Model	53.5	52.3	52.9

As shown in Figure 5, in order to observe data indexes more directly, a line chart was used to compare the *F₁* values of several models in the experimental results. From the overall performance, it is obvious that the method of discovery has the best performance in *F₁*, with *P*, *R*, and *F₁* equal to 53.5, 52.3, and 52.9, respectively. Bi-LSTM showed the worst performance, with an *F₁* value of 49.3.

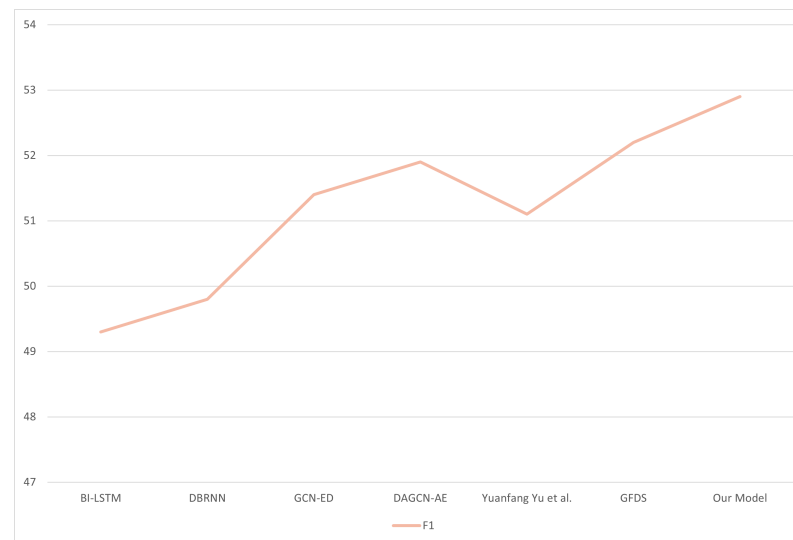


Figure 5. F_1 values of the different models in the datasets [2,3,14–16].

In addition, we found that in the dependency syntactic model, DAGCN had the best performance, with an F_1 score of 51.9, and GCN-ED was second, both higher than DBRNN's F_1 score of 49.8. This is because DBRNN only uses a dependency bridge to enhance the dependency syntactic information, mainly focusing on the grammatical information between words and ignoring information such as the modifier's phrase or verb voice, thus consuming too many computing resources for complex and long sentences. The use of GCN modeling can integrate various information modeling and capture more comprehensive language features. It can also adaptively learn different types of events in different contexts and complex language environments.

However, using Yuanfang Yu et al.'s combined method, the DAGCN's F_1 score decreased by 0.8 compared with the optimal method, and there was no significant improvement compared with DBRNN. After further analysis, it is believed that GCNs are more suitable for complex and diverse open-domain event extraction tasks. At the same time, DAGCNs increase the attention perception and can adaptively weigh different nodes, fine-tuning the features of different nodes and improving the generalization ability of the model. In contrast, the GFDS joint approach effectively integrates semantic and dependency syntactic information. Not only are semantics the basis of the dependency syntactic structure, but sentence dependency is also conducive to learning word-level semantics. In addition, the gating mechanism can be used to selectively update and reset the state to prevent gradients from vanishing or exploding.

In conclusion, this method not only considers the importance of semantic information but also uses semantics to enhance dependency syntactic information and then uses DAGCNs to realize the modeling and automatically strengthen the weight of key nodes. In order to avoid the redundancy of the structural information and dependency syntax information contained in the final layer, the BERT non-adjacent intermediate layer features were used as the input of Bi-LSTM, and the gating mechanism was used to control the flow of information between the different features, thus improving the flexibility of the model.

4.3.2. Ablation Experiments

(1) Influence of different factors on model performance.

In order to analyze the influence of each part of the method and evaluate the contribution of different factors on the model's performance, further ablation experiments were conducted on the dataset to gradually eliminate the following four modules to observe the performance changes, namely the Bi-LSTM final layer, the Bi-LSTM intermediate layer, the DAGCN layer, and the gating mechanism. In particular, since the gating mechanism is bidirectionally weighted, if the Bi-LSTM final layer module is removed, the model only has

a unidirectional input, so the ablation of this part is not gated by default. For the ablation of the DAGCN layer, considering the difference between the features of the BERT output to the middle and final layers, gating can also effectively control the flow of information; therefore, we chose to use a gated dynamic fusion. We used a concatenation layer for feature fusion when the gating mechanisms were eliminated. Table 3 lists the results.

Table 3. Effects of different factors on the model’s performance.

Method	Scheme Matching (%)		
	<i>P</i>	<i>R</i>	<i>F₁</i>
– Bi-LSTM Final Layer	52.9	51.7	52.3
–Bi-LSTM Middle Layer	53.4	52.0	52.7
–DAGCN	53.0	51.2	52.1
–Gating Mechanism	53.1	51.7	52.4

It can be seen from the experimental results that different results are produced by eliminating different layers in the open-domain event extraction task. It was found that the DAGCN layer has a great impact on model performance, while the Bi-LSTM final layer and the gating mechanism have a lesser impact, and the Bi-LSTM intermediate layer has the least influence. Removing the DAGCN layer reduced the model performance by 0.8, again providing reliable validation of the semantically enhanced dependency syntax approach. The elimination of the Bi-LSTM final layer resulted in a 0.6 decrease in model performance, indicating that the Bi-LSTM final layer and gating mechanism have certain effects on the model, but the convolution of the dependency attention diagram is not as important for feature modeling. Since the elimination of the Bi-LSTM final layer does not carry the gating mechanism by default, we ablate the gating mechanism alone, and the experimental results show that there is no significant difference between the model *F₁* values and the elimination of the Bi-LSTM final layer, further demonstrating the effectiveness of the gating mechanism, or, in other words, the necessity of feature fusion.

At the same time, if the Bi-LSTM intermediate layer factor is eliminated, the performance of the model decreases to 52.7, which is relatively mild. This indicates that the middle layer has little effect on the performance of the model. This may be because the feature representation provided by some intermediate layers is similar to that of the final layer, which has a lesser impact on the model performance, while the gating mechanism can dynamically integrate the semantic and dependency syntactic information to provide a richer representation.

(2) The impact of intermediate layer selection on model performance.

In addition, different to the study of Yuanfang Yu et al. [16], this paper conducted an experiment on the open-domain event extraction task. Considering that the data distribution and feature methods of different sub-tasks are different and that the selection of the middle layer will affect the performance of the model, we aimed to select the optimal result to improve the extraction performance. Therefore, reference experiments were carried out to select different intermediate layers. From layers 1 (the final layer), 3, and 4, we selected intervals of 4, 3, and 2, respectively, to reduce. The experimental results are shown in Table 4.

Table 4. Effects of intermediate layer selection on model performance.

Spacing Number	Corresponding Layer	<i>P</i>	<i>R</i>	<i>F₁</i>
5	1, 6, 11	51.6	50.0	50.8
4	1, 5, 9, 11	52.5	51.7	52.1
3	1, 4, 7, 10	53.6	52.2	52.9
3	2, 5, 8, 11	52.5	52.3	52.4
3	3, 6, 9, 12	51.8	51.4	51.6
2	2, 4, 6, 8	50.9	51.5	51.2
2	5, 7, 9, 11	51.2	50.6	50.9
1	13	51.5	49.2	50.3

The experiment proved that selecting a layer spacing of 3 and a layer number of 1, 4, 7, or 10 is the most effective. This is attributed to the joint influence of the intermediate layer spacing and the initial layer. If the interval is too large, the feature correlation learned will be too low, affecting the feature representation. If the interval is too small, the feature learning of the adjacent layers will be very similar, causing redundancy. In the selection of different layers with the same interval of three, the initial layer of one performs very well, considering that the initial layer can contain the most important original features, while features obtained after stacking the layers will lose the key representation, resulting in performance degradation.

5. Conclusions

In this paper, an improved open-domain event extraction method based on a dynamic fusion of semantic and dependency syntax is proposed for open-domain event extraction tasks. Based on a BERT fusion of semantic information and semantically enhanced dependency syntax information, rich feature representations were obtained and input into the neural topic model, solving the problem of existing open-domain event extraction methods, which lack semantic information and fail to capture long-range information. The improvement was made by the following aspects: Firstly, BERT final layer features were output through the Bi-LSTM to obtain a rich semantic representation. Secondly, to avoid semantic and syntactic interference, BERT middle layer features were obtained and input into the Bi-LSTM network. Then, DAGCN was used to increase nodes' attention to information and reduce dependency information noise. Thirdly, the gating mechanism was used to integrate semantic information and semantic-enhanced dependency syntactic information to reasonably control the feature flow to improve the accuracy of event extraction. The results of the main and ablation experiments on the dataset show that the improved method of fusing semantic and dependent syntactic information is very effective for the study of open-domain event extraction tasks.

However, current open-domain extraction methods are inferior to traditional methods in terms of their extraction performance. This means that the proposed method has a lot of room for improvement, and more research and investigations are still needed to improve the accuracy and efficiency of open-domain event extraction. Possible future research directions can start with the following points. First, there should be a deeper understanding of context. The existing open-domain event extraction model still has a poor understanding of context and can only extract information relevant to the text surface. In the future, deeper contextual understanding techniques, such as consciousness and knowledge maps, should enable open-domain event extraction models to more accurately understand the text and extract event information. Second, multi-modal information should be integrated. Future open-domain event extraction models need to be able to process multi-modal information, such as pictures and videos. Since multi-modal information has

a strong correlation with text, integrating multi-modal information and text information can enable a more comprehensive understanding of event information, which is of great significance for guiding information extraction in the future. Third, there is a need for more accurate terminology weighting methods. The parts of speech and semantic expression of different words in a sentence are different. In the process of event extraction, attention should be paid to the words that are most relevant to the semantic meaning of the sentence. Therefore, the weight allocation of different words is also very important and is conducive to improving the accuracy of event extraction. We believe that with the development of deep learning and knowledge-mapping technologies, open-domain event extraction techniques will gradually mature and our understanding of natural language models will become deeper and deeper.

Author Contributions: Conceptualization, L.H. and Q.Z.; methodology, Q.Z. and H.W.; software, Q.Z.; formal analysis, Q.Z.; investigation, Q.Z.; resources, L.H.; data curation, Q.Z.; writing—original draft preparation, Q.Z.; writing—review and editing, Q.Z.; visualization, Q.Z.; supervision, J.D.; project administration, H.W.; funding acquisition, L.H. and H.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China (61972003), Promotion Laboratory and the RD Program of the Beijing Municipal Education Commission (KM202210009002), and supported By the Beijing Urban Governance Research Base of North China University of Technology.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Nguyen, T.H.; Cho, K.; Grishman, R. 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, San Diego, CA, USA, 12–17 June 2016; pp. 300–309.
2. Nguyen, T.; Grishman, R. Graph convolutional networks with argument-aware pooling for event detection. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32.
3. Bai, Y. *Research on Key Technologies of Text-Based Event Extraction*; University of Electronic Science and Technology: Chengdu, China, 2022. [CrossRef]
4. Peng, H.; Li, J.; Song, Y.; Yang, R.; Ranjan, R.; Yu, P.S.; He, L. Streaming social event detection and evolution discovery in heterogeneous information networks. *ACM Trans. Knowl. Discov. Data* **2021**, *15*, 1–33. [CrossRef]
5. De Vroe, S.B.; Guillou, L.; Stanojević, M.; McKenna, N.; Steedman, M. Modality and negation in event extraction. *arXiv* **2021**, arXiv:2109.09393.
6. Wang, R.; Zhou, D.; He, Y. Open event extraction from online text using a generative adversarial network. *arXiv* **2019**, arXiv:1908.09246.
7. Arnulphy, B.; Tannier, X.; Vilnat, A. Automatically generated noun lexicons for event extraction. In Proceedings of the Computational Linguistics and Intelligent Text Processing: 13th International Conference, CICLing 2012, New Delhi, India, 11–17 March 2012; Proceedings, Part II 13; Springer: Berlin/Heidelberg, Germany, 2012; pp. 219–231.
8. Zhou, D.; Chen, L.Y.; He, Y. A simple bayesian modelling approach to event extraction from twitter. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Baltimore, MD, USA, 22–27 June 2014; pp. 700–705.
9. Liu, X.; Huang, H.; Zhang, Y. Open domain event extraction using neural latent variable models. *arXiv* **2019**, arXiv:1906.06947.
10. Chau, M.T.; Esteves, D.; Lehmann, J. A Neural-based model to Predict the Future Natural Gas Market Price through Open-domain Event Extraction. In Proceedings of the CLEOPATRA@ESWC, Heraklion, Crete, Greece, 3 June 2020; pp. 17–31.
11. Yu, J.; Bohnet, B.; Poesio, M. Named entity recognition as dependency parsing. *arXiv* **2020**, arXiv:2005.07150.
12. Kilicoglu, H.; Bergler, S. Syntactic dependency based heuristics for biological event extraction. In Proceedings of the BioNLP 2009 Workshop Companion Volume for Shared Task, Boulder, CO, USA, 5 June 2009; pp. 119–127.
13. Manning, C.D.; Surdeanu, M.; Bauer, J.; Finkel, J.R.; Bethard, S.; McClosky, D. The Stanford CoreNLP natural language processing toolkit. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, Baltimore, MD, USA, 23–24 June 2014; pp. 55–60.
14. Sha, L.; Qian, F.; Chang, B.; Sui, Z. Jointly extracting event triggers and arguments by dependency-bridge RNN and tensor-based argument interaction. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32.

15. Li, J.; Yu, C.; Hong, J.W. *An Event Detection Method Using Gating Mechanism to Fuse Dependency and Semantic Information*; Soochow University: Suzhou, China, 2020; Volume 34, pp. 51–60.
16. Yu, Y. *Study on Event Extraction based on Dependency Syntax and Role Knowledge*; Huazhong Normal University: Wuhan, China, 2022. . [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.