applied sciences

MDPI

Article

Robustness Analysis on Graph Neural Networks Model for Event Detection

Hui Wei, Hanqing Zhu, Jibing Wu *, Kaiming Xiao and Hongbin Huang

Laboratory for Big Data and Decision, National University of Defense Technology, Changsha 410073, China
* Correspondence: wujibing@nudt.edu.cn

Abstract: Event Detection (ED), which aims to identify trigger words from the given text and classify them into corresponding event types, is an important task in Natural Language Processing (NLP); it contributes to several downstream tasks and is beneficial for many real-world applications. Most of the current SOTA (state-of-the-art) models for ED are based on Graph Neural Networks (GNN). However, a few studies focus on the issue of GNN-based ED models' robustness towards text adversarial attacks, which is a challenge in practical applications of EDs that needs to be solved urgently. In this paper, we first propose a robustness analysis framework for an ED model. Using this framework, we can evaluate the robustness of the ED model with various adversarial data. To improve the robustness of the GNN-based ED model, we propose a new multi-order distance representation method and an edge representation update method based on attention weights, then design an innovative model named A-MDL-EEGCN. Extensive experiments illustrate that the proposed model can achieve better performance than other models both on original data and various adversarial data. The comprehensive robustness analysis according to experimental results in this paper brings new insights into the evaluation and design of a robust ED model.

Keywords: robustness; graph neural networks; event detection; multi-order distance

check for updates

Citation: Wei, H.; Zhu, H.; Wu, J.; Xiao, K.; Huang, H. Robustness Analysis on Graph Neural Networks Model for Event Detection. *Appl. Sci.* **2022**, *12*, 10825. https://doi.org/10.3390/app122110825

Academic Editor: Eui-Nam Huh

Received: 8 September 2022
Accepted: 21 October 2022
Published: 25 October 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 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

Event Detection (ED) aims to identify trigger words from a given text and classify them into corresponding event types. As shown in Figure 1, an ED model aims to identify **destroyed** in S1 as an Attack trigger word and **fired** in S2 as an EndPosition trigger word. As an important task in Natural Language Processing (NLP), ED contributes to Event Argument Extraction [1] and Event–Event Relation Extraction [2,3] , and it is beneficial for real-world applications, including Automatic Text Summarization [4], Information Retrieval (IR) [5], and Question Answering (QA) [6].

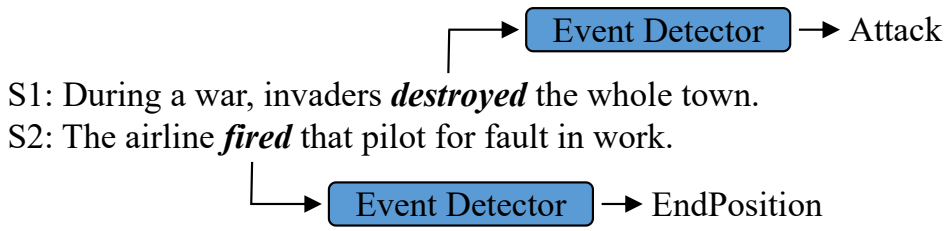

Figure 1. Example of Event Detection.

Traditional models for ED are mainly based on machine learning methods [7–10]. However, the performance of them was heavily dependent on manually selected features. With the improvement of deep learning theory and technology in recent years, deep learning has proven to be more suitable for NLP tasks [11–13]; therefore, more and more researchers use deep learning methods to perform ED [14–17]. Recently, more research

has focused on how to convert a text sequence into graph-structured data to incorporate rich semantic information and then introduce Graph Neural Networks (GNN) into the ED task [18–22], and the current SOTA (state-of-the-art) models for ED are based on GNN.

Most of the above models are based on default and perfect dataset assumption, i.e., the label quality is high, the noise is low, and the distribution is balanced; hence, it is expected that patterns learned from such a dataset can be generalized. However, a real dataset that does not satisfy the completeness tends to result in a trained model that contains the same biases as the training data [23]. The research on ED mainly focuses on the construction of a model and evaluates the model performance according to the metrics on high-quality datasets, such as Precision, Recall, and F1-score, which may lead to an overestimation of the model's ability [24].

In the field of NLP, the robustness of a model is apparently an essential indicator, and a model that has strong adaptability to various input texts is desired for different applications in which their inputs are not always perfect. Although some researchers have begun to pay attention to the robustness of NLP models [25–30], there are few studies on ED models. The issue of ED models' robustness is a challenge for both theoretical research and the practical application of ED, which needs to be solved urgently.

Differently from usual studies that propose a robust model for one specific instance among various kinds of data contamination or imperfections (e.g., modeling the adversarial data or adopting any adversarial optimization objective), our work follows a totally different route toward more general robustness. The main idea is to introduce an endogenous, robust enhancement mechanism rather than propose thousands of over-specific models or methods in isolation. We first propose a Robustness Analysis Framework on an ED Model to evaluate the performance of the ED model facing various text transformations and subpopulations, based on which we can comprehensively analyze the ED model's robustness. To improve the robustness of the GNN-based ED model, we then propose a new multi-order distance representation method (for a syntactic dependency graph of input sentence) to better capture associations between long-distance words and an edge representation update method based on attention weights to better distinguish the importance of different edge types in edge update. The effectiveness of the two methods is verified by extensive experiments.

The main contributions of this paper are:

- In the absence of current research on the robustness of ED models, we propose a Robustness Analysis Framework on an ED Model that facilitates the comprehensive analysis of the ED model's robustness.
- We propose a new multi-order distance representation method and an edge representation update method based on attention weights to enhance EE-GCN, then design an innovative GNN-based ED model named A-MDL-EEGCN. Our experiments illustrate that the performance of this model is better than that of the previously proposed GNN-based ED models on the ACE2005 dataset, especially when adversarial data exists.
- Using the robustness analysis framework on the ED model, we perform extensive experiments to evaluate the performance of several GNN-based ED models, and the comprehensive robustness analysis according to experimental results brings new insights to the evaluation and design of robust ED models.

2. Related Work

2.1. Event Detection

Early on, the research on ED mainly adopted machine learning methods that through feature engineering use statistic and linguistic features such as N-grams, part of speech, etc. [7–10]. However, feature engineering greatly determines the overall performance of the ED model, and also has high requirements for human resources and expertise.

The most prominent advantage of deep learning over machine learning is learning effective features from data through multi-layer neural networks [31]. With the gradual

improvement in deep learning theory and technology in recent years, deep learning has proven to be more suitable for NLP tasks [11–13]; therefore, more and more researchers use deep learning methods to perform ED. Convolutional Neural Networks (CNN) [14] were applied to capture the semantic associations between continuous words through convolution operation; thereby candidate trigger word could aggregate the contextual information. The dynamic multi-pooling strategy [15] was used to model various semantic associations and thus enhanced CNN. Nguyen et al. considered that general CNNs are poor at fusing discontinuous words, then they proposed the skip-grams method [16] to make up for this deficiency. Ghaeini et al. constructed a Recurrent Neural Networks (RNN) model for the first time [17] to detect multi-word events.

Recently, more research has focused on how to convert a text sequence into graph-structured data to incorporate rich semantic information, and then GNN was introduced into the NLP tasks. Nguyen et al. first leveraged Graph Convolutional Networks (GCN) [18] to perform ED through a syntactic dependency tree and achieved remarkable model performance. Liu et al. introduced a self-attention mechanism and highway network structure into GCN [19] to improve the performance of ED. MOGANED utilized multi-order syntactic dependency to aggregate each order word vector representation through the attention mechanism [20], further improving the performance of GCN. Cui et al. proposed the learning of the embedding vectors for the edges of the syntactic dependency graph by the node update module and edge update module [21], and achieved the SOTA effect of GCN. Lai et al. achieved an improvement in the GCN effect by using trigger word filters [22] to reduce the influence of irrelevant noise between adjacent words.

2.2. Robustness Research in Natural Language Processing

Papernot et al. first studied how to design adversarial text sequences for RNN [25]. Alzantot et al. designed a heuristic optimization algorithm [26] to semantically and syntactically generate similar adversarial text samples. A greedy algorithm named PWWS [27] was proposed for generating adversarial text samples that preserve lexical correctness, grammatical correctness, and semantic similarity. TextAttack is a platform that can use adversarial attacks, data augmentation, and adversarial training for NLP tasks [28], which only needs to define an objective function, a set of constraints, a text transformation, and a search method to reproduce the previously proposed or customed text attack algorithm for generating high-quality adversarial text samples. OpenAttack is different from and complementary to TextAttack in supporting all attacks, multilinguality, and parallel processing [29]. TextFlint is a multilingual robustness evaluation platform for NLP tasks [30] that not only incorporates text transformations, adversarial attacks, subpopulations, and their combinations but also automatically generates visualization reporst, which facilitates a comprehensive robustness analysis.

The issue of the ED model's robustness is of great practical significance and needs to be solved, but there are few studies on it. Lu et al. proposed a Δ-representation learning approach [32] distinguish ambiguous trigger words and detect unseen/sparse trigger words by effectively decoupling, learning, and fusing alterable incremental parts for event representation, instead of learning a single comprehensive representation. Although Lu et al. considered the ambiguity and sparsity of the input text, they ignored the crafted adversarial text, making the ED model not very robust. Liu et al. proposed a training paradigm called Context-Selective Mask Generalization [33], which improved the model's robustness against adversarial attacks, out-of-vocabulary (OOV) trigger words, and ambiguous trigger words. However, Liu et al. just utilized the algorithm of Alzantot [26] to generate adversarial samples perturbing only the trigger words or all words to evaluate the robustness of the ED model without considering different types of text transformation and subpopulation, which makes the analysis of the model's robustness not comprehensive enough.

3. Robustness Analysis Framework on Event Detection Model

This paper regards ED as a sequence labeling task. The input of the task is a sequence of natural text, each word in the text is regarded as a token, and the sequence of event types corresponding to each token is used as the output. Formally, given an input sequence $S = (w_1, w_2, \ldots, w_n)$ containing n tokens, the corresponding sequence of event types is $ET = (et_1, et_2, \ldots, et_n)$, where the event types are annotated by 'BIO' schema.

TextFlint can only perform the robustness analysis of specific NLP tasks such as Named Entity Recognition (NER), Relation Extraction (RE), Part of Speech Tagging (POST), and Sentiment Analysis (SA). Therefore, we use the functions provided by TextFlint to build a Robustness Analysis Framework on an ED model, as shown in Figure 2. This framework utilizes TextFlint to generate adversarial data, including transformed data and data subpopulations from the original data, and compares the performance of the ED model to the original data, based on which we can comprehensively analyze the model's robustness. The text transformations and subpopulations used are described below.

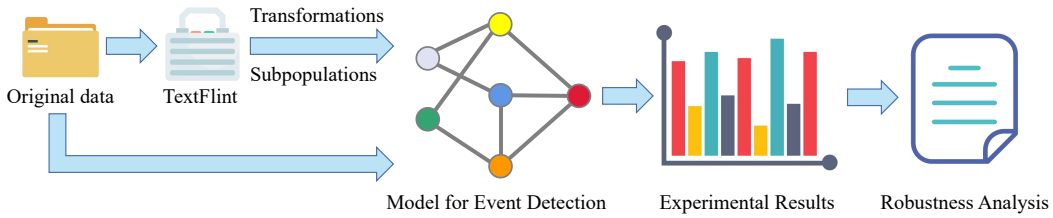

Figure 2. The architecture of the robustness framework on an ED model .

3.1. Text Transformations

Keyboard, Ocr, SpellingError, Tense, Typos, and SwapSyn were selected from the universal text transforms. Meanwhile, since A-MDL-EEGCN, EE-GCN, and MOGANED all take entity types into input, EntTypos was selected from the NER-specific text transformations. The descriptions of the above seven text transformations are shown in Table 1.

Table 1. Text transformations and corresponding descriptions.

Transformations	Descriptions
Keyboard	Simulates the errors of how people type words with the use of keyboard.
Ocr	Simulates Ocr error by random values.
SpellingError	Simulate possible mistakes in the spelling of words.
Tense	Transforms all verb tenses in a sentence.
Typos	Randomly inserts, deletes, and swaps a letter within one word.
SwapSyn	Replaces one word with its synonym provided by WordNet [34].
EntTypos	Applies Typos for words with entity type label.

3.2. Subpopulations

Due to the length of each input text not being exactly equal, the sequence labeling model usually sets a maximum text length to fill short text (by placeholder) and truncate long text in order to output a prediction sequence with the same length. Therefore, the length subpopulation was chosen to screen the original data based on text length for generating data subpopulations.

In addition, the Perplexity of the GPT-2 language model [35] was chosen to screen the original data to generate data subpopulations; its formula is as follows:

$$\text{Perplexity}(S) = \Big(p(w_1, w_2, \ldots, w_n)\Big)^{-\frac{1}{n}}, \tag{1}$$

where $p(w_1, w_2, \ldots, w_n)$ is the probability that GPT-2 language model generates text sequence $S = (w_1, w_2, \ldots, w_n)$.

In short, the higher Perplexity of S, the weak its plausibility; thus Perplexity could roughly embody the plausibility of S.

4. Model

For the given input sequence $S = (w_1, w_2, \ldots, w_n)$, we vectorize each w_i to $\mathbf{x}_i = [\mathbf{w}_i, \mathbf{e}_i] \in \mathbb{R}^{(d_w+d_e)}$, where $\mathbf{w}_i$ (pre-trained on the NYT corpus by the skip-gram method) and $\mathbf{e}_i$ (entity type is annotated by 'BIO') represent the word-embedding vector and entity-type embedding vector of w_i, d_w and d_e are the dimension of above vectors, respectively. Then we feed $[\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n] \in \mathbb{R}^{n\times(d_w+d_e)}$ into a Bi-LSTM with a hidden size of $d_l/2$, $\overrightarrow{\mathbf{h}_i}, \overleftarrow{\mathbf{h}_i} \in \mathbb{R}^{d_l/2}$ are the forward and reverse hidden states of $\mathbf{x}_i$. Finally, $\mathbf{h}_i = [\overrightarrow{\mathbf{h}_i}, \overleftarrow{\mathbf{h}_i}] \in \mathbb{R}^{d_l}$ was used to represent each token w_i.

Meanwhile, we conduct syntactic dependency parsing on the input sequence S. Figure 3 shows the syntactic dependency parsing of S2 in Figure 1. By taking words as nodes and dependencies as edges, we obtain a syntactic dependency graph (adjacency matrix).

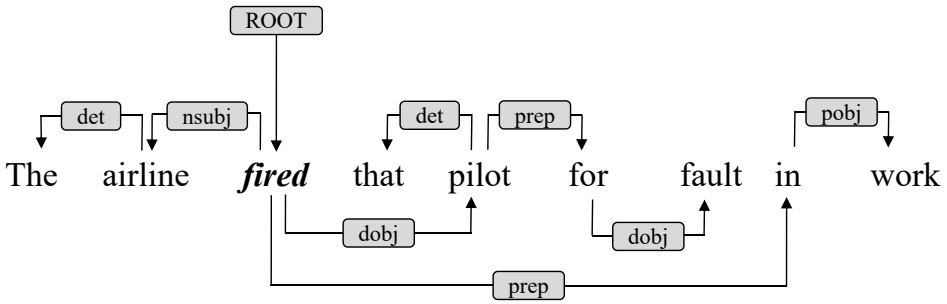

Figure 3. Example of syntactic dependency parsing.

4.1. Edge-Enhanced Graph Convolution Networks

Due to previous GNN-based ED models usually ignoring dependency label information, which conveys rich and useful linguistic knowledge for ED, Cui et al. proposed EE-GCN [21], which embeds the edges of the syntactic dependency graph into a vector space to obtain an edge representation tensor $\mathbf{EM} = [\mathbf{em}_{i,j,k}] \in \mathbb{R}^{n\times n\times p}$, where $\mathbf{em}_{i,j} \in \mathbb{R}^p$ is the vector representation of the corresponding edge in the syntactic dependency graph, it contains more semantic information than 0 or 1 of the traditional adjacency matrix.

Denoting $\mathbf{H}^0 = (\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_n) \in \mathbb{R}^{n\times d_l}$. After each node (word) representation was converted to the dimension of d_g, $\mathbf{H}^0$ is the input state of layer 1 in EE-GCN. The vector of each node in layer $l \in [1, L]$ updates by aggregating the information from its neighbor nodes through the edge representation tensor, so the output state of each layer l ($\mathbf{H}^l \in \mathbb{R}^{n\times d_g}$) is as follows:

$$\mathbf{H}^l = \sigma(\text{Pool}(\mathbf{H}^l_1, \mathbf{H}^l_2, \ldots \mathbf{H}^l_p)), \tag{2}$$

Specifically, the aggregation is conducted channel by channel in the adjacency tensor as follows:

$$\mathbf{H}^l_k = \mathbf{EM}^{l-1}_{:,:,k}\mathbf{H}^{l-1}_k\mathbf{W}_N, k \in [1, p], \tag{3}$$

where Pool is the mean-pooling operation to compress information from all channels, $\mathbf{W}_N \in \mathbb{R}^{d_g\times d_g}$ is a learnable parameter, and σ is the ReLU activation function.

The vector of each edge in layer l updates as follows:

$$\mathbf{em}^l_{i,j} = \mathbf{W}_E[\mathbf{em}^{l-1}_{i,j} \oplus \mathbf{h}^l_i \oplus \mathbf{h}^l_j], i, j \in [1, n], \tag{4}$$

where $\mathbf{W}_E \in \mathbb{R}^{(2*d_g+p)\times p}$ is a learnable parameter and $\oplus$ is the join operation.

We feed the final representation of each word (node) $\mathbf{h}_i^L$ to a fully-connected network, which is followed by a softmax function to compute the probability distribution over all event types as follows:

$$p(y_i|\mathbf{h}_i^L) = \text{softmax}(\mathbf{W}_C\mathbf{h}_i^L + \mathbf{b}_C), \quad (5)$$

where $\mathbf{W}_C$ maps the word representation $\mathbf{h}_i^L$ to the feature score for each event type, and $\mathbf{b}_C$ is a bias term. The event label with the largest probability is chosen as the classification result.

The bias loss function is used to enhance the influence of event labels during training:

$$J(\theta) = -\sum_{i=1}^{N_s}\sum_{j=1}^{n_i} \log p(y_j^t|s_i,\theta)\cdot I(O) + \alpha \log p(y_j^t|s_i,\theta)\cdot(1-I(O)), \quad (6)$$

where N_s is the number of sentences, n_i is the number of words in sentence s_i, and y_j^t is the ground-truth event label of words; $I(O)$ equals 1 if the event type of the word is 'O', otherwise 0; α is the bias weight large than 1.

4.2. Enhancement of EE-GCN

According to the statistical results in the ACE2005 dataset, about 51% of event-related words need at least two hops to get to the corresponding trigger words in the syntactic dependency graph; we propose a new multi-order distance representation method, i.e., introducing multi-order distance labels, which contributes to capturing the associations between long-distance words so that it enhances the contextual awareness of trigger words, especially in long sentences. "nsubj" (nominal subject), "dobj" (direct object), and "nmod" (noun compound modifier) make up 32.2% of trigger word-related dependency labels, we propose an edge representation update method based on attention weights to better distinguish the importance of different edge types in an edge update when multi-order distance labels are introduced.

Differently from usual studies, which propose a robust model for one specific instance among various kinds of data contamination or imperfections, our work follows a totally different route toward more general robustness. The main idea is to introduce an endogenous robust enhancement mechanism rather than propose thousands of over-specific models or methods in isolation. Hence, inspired by the way humans read and understand natural language, we believe that comprehensive comprehension of multi-order distance words is an intrinsic mode of human understanding of language, which helps humans fully understand the meaning of sentences. Even if there are a few mistakes (i.e., various kinds of data contamination or imperfections) in the sentence, it will not affect their correct understanding. Therefore, drawing on this natural endogenous robustness mechanism, we propose the multi-order distance representation method (similarly in the design of attention weights) in a targeted manner to achieve more general robustness.

The above two methods are utilized to enhance EE-GCN, thereby, we design a new GNN-based ED model named A-MDL-EEGCN, and its architecture is shown in Figure 4. A-MDL-EEGCN makes up for the defect of EE-GCN that does not consider multi-order distance and the defect of MOGANED that ignores dependency labels information [36].

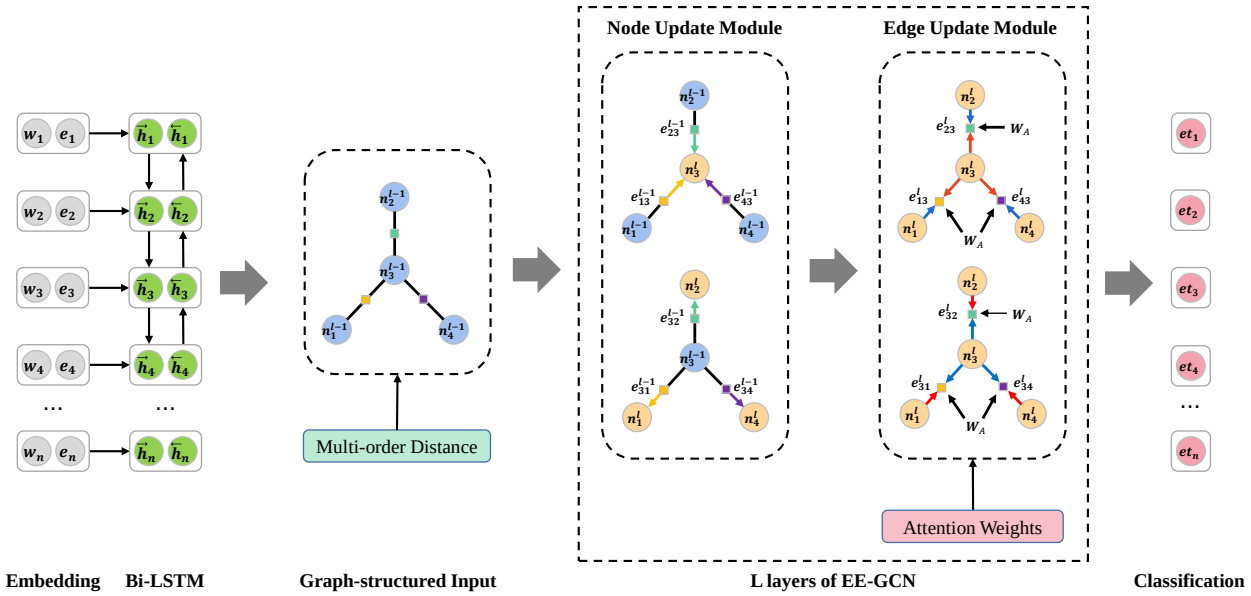

Figure 4. The architecture of A-MDL-EEGCN.

4.2.1. Multi-Order Distance Representation Method

Distance refers to the syntactic distance (least hops) between two words (nodes) in the syntactic dependency graph. As shown in Figure 5, the solid line and dashed line represent 1-order distance and multi-order distance, respectively.

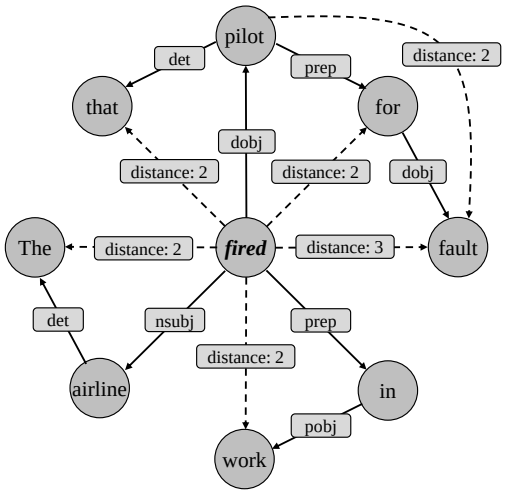

Figure 5. Syntactic dependency graph with multi-order distance.

MOGANED hierarchically introduced multi-order distance [20]; that is, each order word vector is calculated by the corresponding GCN layer and aggregates to be the final vector representation by the attention module. However, this method requires considerable computation. Therefore, we propose a new multi-order distance representation method, i.e., label 2-order distance and 3-order distance to "distance: 2" and "distance: 3", respectively. In this way, the embedding vectors of these edges can participate in the node and edge update of EE-GCN instead of separate calculation. Thus the EE-GCN is able to better capture associations between long-distance words.

4.2.2. Edge Representation Update Method Based on Attention Weights

Separately introducing the above method into EE-GCN led to performance degradation, thus we considered that the edge representation update method of EE-GCN had

difficulty distinguishing the importance of different edge types in edge updates when multi-order distance labels are introduced. For example, as shown in Figure 5, "pilot" is the object in the "EndPosition" event and "fired" is a trigger word, so the edge "dobj" between "fired" and "pilot" should contain important semantic information and accordingly have higher weights after the edge update.

In short, each edge should have a weight associated with its head and tail node when updated. Therefore, an edge representation update method based on attention weights is proposed in this paper. Each vector representation of an edge is updated according to the weight score, which is calculated from the attention aggregation of its head and tail node vector representation; this method can be mathematically defined as follows:

$$\mathbf{em}_{i,j}^{l} = \mathbf{em}_{i,j}^{l} \cdot \mathbf{W}_A \cdot [\mathbf{h}_i^l \oplus \mathbf{h}_j^l], i, j \in [1, n] \tag{7}$$

where $\mathbf{W}_A \in \mathbb{R}^{(2*d_g)\times 1}$ is a learnable parameter.

5. Experiments

5.1. Implementation Details

The experimental dataset is ACE2005, and data preprocessing is the same as MO-GANED and EE-GCN, including syntactic dependency parsing (by using the Stanford CoreNLP toolkit [37]) and data split.

Precision (P), Recall (R), and F1-score (F1) are used as metrics. For the sequence labeling task, positive and negative refers to words with non-"O" and "O" labels, respectively. The mathematical definition of the above metrics as follows:

$$\mathrm{P} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \times 100\% \tag{8}$$

$$\mathrm{R} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \times 100\% \tag{9}$$

$$\mathrm{F}_1 = \frac{2 \times \mathrm{P} \times \mathrm{R}}{\mathrm{P} + \mathrm{R}} \tag{10}$$

where TP, FP, and FN denote true positives, false positives, and false negatives, respectively.

Obviously, P indicates the percentage of predicted positives that are true positives, while R indicates the percentage of actual positives that are true positives, F1 is the harmonic average of P and R.

In order to fairly compare the performance, each model's hyper-parameters are the parameters that achieve the best performance on the original data. The hyper-parameters of A-MDL-EEGCN are listed in Table 2.

Table 2. Hyper-parameters of A-MDL-EEGCN.

Hyper-Parameters	Values
Dimension of word vectors (d_w)	100
Dimension of entity types vectors (d_e)	50
Dimension of edge labels vectors (p)	50
Dimension of Bi-LSTM ($d_l/2$)	100
Dimension of GCN (d_g)	150
Layers of GCN (L)	2
Learning rate	0.001
Optimizer	Adam [38]
Bias weight of loss function (α)	5
Batch size	30
Epoch	100
Maximum text length	50

5.2. Model Performance on the Original Data

The experimental results on the original data are shown in Table 3, where MDL-EEGCN refers to EE-GCN with the separate introduction of proposed multi-order distance labels (MDL), and A-EEGCN refers to EE-GCN enhanced by the edge representation update method based on attention weights.

Table 3. Performance of each model on the original data.

Model	P	R	F1
GCN-ED [18]	77.9	68.8	73.1
JMEE [19]	76.3	71.3	73.7
MOGANED [20]	**79.5**	72.3	75.7
GatedGCN [22]	78.8	76.3	77.6
EE-GCN [21]	76.7	78.6	77.6
MDL-EEGCN	78.9	75.6	77.2
A-EEGCN	77.6	78.4	78.0
A-MDL-EEGCN	78.2	**78.7**	**78.4**

Bold indicates the highest value.

The results show that the A-MDL-EEGCN we proposed is better than MOGANED and GatedGCN on R and F1 and than GCN-ED, JMEE, and EE-GCN on all metrics, illustrating A-MDL-EEGCN can achieve better performance than the previously proposed GNN-based ED models. We further analyze the effect of the two enhancement methods.

Although MDL-EEGCN is better than MOGANED on R and F1, it is worse than EE-GCN on R and F1. By checking the edge representation tensor **EM** of MDL-EEGCN, we find that this is because the l2 norm (regarded as the association score of word pair) of the embedding vector of edge "distance: 2" is larger than some 1-order distance edges. Thus we believe that when new edge types are introduced into EE-GCN, it is difficult for the edge representation update method of EE-GCN to distinguish the importance of different edge types in the edge update, but instead dilutes the original semantic, then degrades the performance. A-EEGCN improves on P and F1 compared to EE-GCN, confirming the effectiveness of the edge representation update method based on attention weights. A-MDL-EEGCN outperforms A-EEGCN and MDL-EEGCN on R and F1, meanwhile, its **EM** accurately embodies the difference of each order distance edge importance, demonstrating the positive effect of MDL and the necessity of combining the two methods we propose in this paper.

5.3. Model Performance on the Adversarial Data

We did no additional tuning to the models so that the models' robustness can be compared fairly and the effectiveness of the two methods proposed in this paper can be validated.

For text transformations, we set Tense to transform all verb tenses in the input text, SwapSyn to replace each word in the input text with probability = 0.5, and other text transformations to transform each word with probability = 0.3.

For subpopulations, since the maximum text length of GNN-based ED models we use is 50, we screen the original data with text lengths, then generate length ≤ 50 and length > 50 to evaluate the effect of padding and truncating input text on the model performance. Meanwhile, we use the Perplexity of the GPT-2 language model as a metric to sort the original data, then generate Perplexity-0-50% and Perplexity-0-20% to evaluate the model performance on input text with high Perplexity.

The data generated by text transformations and subpopulations are collectively called adversarial data. Notice that due to the randomness of text transformations we performed more than 10 runs (conducting text transformations by setting different random seeds) to test the performance of models. The t-test (significance level $p = 0.05$) indicates that there was no statistically significant difference among these 10 test runs , so that we only report one test run result. Table 4 shows the performance of A-MDL-EEGCN, EE-GCN,

and MOGANED on each adversarial data, i.e., the robustness evaluation results of these GNN-based ED models.

In most cases, it is consistent with the performance on the original data that A-MDL-EEGCN performs best on R while MOGANED on P. The definitions of P and R are both critical, but they are generally contradictory. Therefore, we used composite metric F1 to embody the model's robustness and conducted the following analysis.

Table 4. Robustness evaluation results.

Adversarial Data	A-MDL-EEGCN			EE-GCN			MOGANED		
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
Keyboard	**72.1**	58.1	64.3	70.9	**59.5**	**64.7**	70.8	48.4	57.5
Ocr	69.6	**52.6**	**59.9**	**73.2**	47.9	57.9	71.5	42.7	53.5
SpellingError	71.1	**56.7**	**63.1**	**73.4**	55.1	62.9	69.2	47.5	56.3
Typos	71.7	**49.9**	**58.8**	71.0	47.7	57.0	**71.9**	40.7	52.0
EntTypos	**74.5**	**77.5**	**75.8**	71.8	75.9	73.8	71.8	65.3	68.4
Tense	71.1	**77.2**	**74.0**	71.3	74.9	73.1	**72.2**	63.9	67.8
SwapSyn	73.2	**72.4**	**72.8**	69.6	68.5	69.1	**73.7**	60.0	66.2
Tense + Typos	**70.5**	**51.9**	**59.8**	69.6	49.0	57.5	66.2	39.5	49.5
SwapSyn + Typos	68.9	**41.9**	**52.1**	**70.3**	40.3	51.2	67.4	34.3	45.5
Length ≤ 50	79.2	**79.1**	**79.1**	78.0	78.6	78.3	**79.7**	72.6	76.1
Length > 50	63.6	**71.8**	**67.5**	59.0	59.0	59.0	**73.4**	56.3	63.7
Perplexity-0-50%	69.8	**77.0**	**73.2**	68.8	74.4	71.5	**73.6**	66.1	69.7
Perplexity-0-20%	65.8	**75.0**	**70.1**	67.0	70.9	68.9	**69.6**	64.9	67.1

Bold indicates the highest value.

5.3.1. Model Robustness to Character-Level Transformations

Keyboard, Ocr, SpellingError, Typos, and EntTypos all transform one or several characters in a word, which are character-level transformations. The experimental results show that the robustness of models to EntTypos is significantly ($p = 0.05$) stronger than that of the other four. It is obvious that EntTypos is only for words with entity labels so that it causes less perturbation to the original sentence than other character-level transformations. Further, we analyze the robustness of the models to four other text transformations one by one:

- The perturbation caused by Typos is irregular, and the transformed words will almost certainly be OOV words, so the robustness of models to Typos is very weak.
- Although Ocr simulates possible errors in reality, the robustness of the model to it is also weak. We believe that because the corpus for training word vectors is manually typed rather than recognized from pictures, errors caused by Ocr rarely appear in the corpus.
- SpellingError and Keyboard simulate errors that may be caused by humans and appear in the corpus, so models are more robust to these two text transformations than the other two.

It can be seen from the above analysis that the robustness of the GNN-based ED models to character-level transformations is related to the training corpus. Although these models use the same pre-trained word vectors, A-MDL-EEGCN and EE-GCN are more robust to character-level transformations than MOGANED. We infer that since MOGANED only considers adjacency instead of dependency labels, it is more sensitive to noise from transformed words.

5.3.2. Model Robustness to Word-Level Transformations

Both Tense and SwapSyn are word-level transformations because they transform one word into another. The experimental results show that the model's robustness to Tense and SwapSyn are both relatively strong, while the former is a little stronger than the latter. We have conducted the following analysis:

- Transforming all verb tenses basically does not change the meaning of the sentence, and the semantic difference between verbs in different tenses is small; the corresponding word vectors should be very similar, thus Tense causes little perturbation to the original sentence.
- Replacing words with synonyms slightly changes the meaning of the sentence (e.g., the degree of emotion); although word vectors of synonyms should also be similar, SwapSyn causes perturbation to the original sentence a little more than Tense.

It can be seen from the above analysis that the GNN-based ED model can cope with the slight change in lexical features well; that is, the model can handle sentences with different expressions but nearly the same meaning. A-MDL-EEGCN is more robust to word-level transformations than EE-GCN and MOGANED.

5.3.3. Model Robustness to Combining Text Transforms

We combine Typos, the character-level transformation that has the greatest effect on model performance, with word-level transformations. As the character-level transformation will affect the recognition of words, we first conduct SwapSyn (Tense) on the original sentence and then conduct Typos, which is called SwapSyn + Typos (Tense + Typos). The experimental results show that the combination of text transformations will further degrade the model's performance, which suggests that we can create more new combinations of text transformations to evaluate the robustness of the model more comprehensively.

5.3.4. Model Robustness to Data Subpopulations

The experimental results show that the performance of models upgrade where the length ≤ 50, while degrade where length > 50. The reason is obvious:

- In the program, the model masks the filled placeholder (padding) at the end of the input sequence. When reading, humans also ignore meaningless symbols at the end of sentences. Therefore, a short sentence filled with placeholders still retains the original meaning.
- On the contrary, truncation affects the structural and semantic integrity of a long sentence (i.e., making the sentence incomplete and difficult to understand for both humans and machines); thus the important information may be lost.

Most of the text in the original data are short sentences so that the performance of each model on length ≤ 50 is comparable to that of the original data. However, many associations between long-distance words exist in a long sentence, A-MDL-EEGCN and MOGANED significantly ($p = 0.05$) outperform EE-GCN on length > 50, illustrating that it is essential to capture these associations in long sentences for ED. Moreover, the time consumption in the training of MOGANED (about 1000 s per epoch) is much longer than that of A-MDL-EEGCN and EE-GCN (about 6 s per epoch) in our experimental environment, illustrating that A-MDL-EEGCN is efficient and effective.

The model's performance on Perplexity-0-20% is worse than that on Perplexity-0-50%, and the model's performance on Perplexity-0-50% is worse than that on the original data, indicating that the Perplexity of GPT-2 could measure the quality of the input text and the quality of the input text affects ED model performance. Perplexity-0-20% is the top 20% input texts in terms of Perplexity, and the models' F1 on it decreases by not more than 10, demonstrating that the GNN-based ED models are robust in regards to texts with high Perplexity. Moreover, more metrics than just Perplexity are needed to measure the quality of the input text to evaluate the ability of the ED model to detect events represented by low-quality text.

6. Conclusions

To study the robustness of ED models, we first proposed a Robustness Analysis Framework on an ED model to evaluate the performance of the ED model facing various text transformations and subpopulations. Further, we proposed a new multi-order distance rep-

resentation method and an edge representation update method based on attention weights to enhance GNN, then designed an innovative GNN-based ED model, A-MDL-EEGCN. The main idea is to introduce an endogenous robust enhancement mechanism rather than propose thousands of over-specific models or methods in isolation. Finally, we used the Robustness Analysis Framework on an ED model to perform extensive experiments, i.e., a comprehensive robustness evaluation for several GNN-based ED models and analyzed the reasons based on the experimental results for the difference in the model's robustness on different adversarial data.

Notably, our proposed model showed a general superiority over other GNN-based models, especially when different adversarial data exist. The comprehensive robustness analysis, according to the experimental results, brings new insights into the evaluation and design of robust ED models.

This study had the following limitations:

1. This paper only focuses on GNN-based ED models, while other models are also worthy of in-depth study and analysis. We expect more novel and robust model structures to emerge in the future.
2. Text transformations and subpopulations contained in the Robustness Analysis Framework on an ED model were limited, and we encourage future studies focused on ED model robustness to consider more types (or combinations) of adversarial text attacks.

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

Funding: This research was funded by the Provincial Natural Science Foundation of Hunan, grant number 2019JJ50726.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. These data can be found here: https://catalog.ldc.upenn.edu/LDC2006T06 (accessed on 15 February 2006).

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

References

1. Doddington, G.R.; Mitchell, A.; Przybocki, M.A.; Ramshaw, L.A.; Strassel, S.M.; Weischedel, R.M. The automatic content extraction (ace) program-tasks, data, and evaluation. In Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC), Lisbon, Portugal, 26–28 May 2004; pp. 837–840.
2. Han, R.; Zhou, Y.; Peng, N. Domain Knowledge Empowered Structured Neural Net for End-to-End Event Temporal Relation Extraction. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020; pp. 5717–5729.
3. Zuo, X.; Cao, P.; Chen, Y.; Liu, K.; Zhao, J.; Peng, W.; Chen, Y. LearnDA: Learnable Knowledge-Guided Data Augmentation for Event Causality Identification. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Online, 1–6 August 2021; Volume 1: Long Papers, pp. 3558–3571.
4. Marujo, L.; Ribeiro, R.; Gershman, A.; de Matos, D.M.; Neto, J.P.; Carbonell, J. Event-based summarization using a centrality-as-relevance model. *Knowl. Inf. Syst.* **2017**, *50*, 945–968. [CrossRef]
5. Campos, R.; Dias, G.; Jorge, A.M.; Jatowt, A. Survey of temporal information retrieval and related applications. *ACM Comput. Surv. (CSUR)* **2014**, *47*, 1–41. [CrossRef]
6. Wang, J.; Jatowt, A.; Färber, M.; Yoshikawa, M. Improving question answering for event-focused questions in temporal collections of news articles. *Inf. Retr. J.* **2021**, *24*, 29–54. [CrossRef]
7. Ahn, D. The stages of event extraction. In Proceedings of the Workshop on Annotating and Reasoning about Time and Events, Sydney, Australia, 23 July 2006; pp. 1–8.
8. Ji, H.; Grishman, R. Refining Event Extraction through Cross-Document Inference. In Proceedings of the ACL-08: HLT, Columbus, OH, USA, 15–20 June 2008; pp. 254–262.

9. Liao, S.; Grishman, R. Using Document Level Cross-Event Inference to Improve Event Extraction. In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Uppsala, Sweden, 11–16 July 2010; pp. 789–797.
10. Hong, Y.; Zhang, J.; Ma, B.; Yao, J.; Zhou, G.; Zhu, Q. Using Cross-Entity Inference to Improve Event Extraction. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Portland, OR, USA, 19–24 June 2011; pp. 1127–1136.
11. Bengio, Y.; Ducharme, R.; Vincent, P. A neural probabilistic language model. In Proceedings of the Advances in Neural Information Processing Systems 13 (NIPS 2000), Denver, CO, USA, 1 January 2000.
12. Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; Kuksa, P. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.* **2011**, *12*, 2493–2537.
13. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed representations of words and phrases and their compositionality. In Proceedings of the Advances in Neural Information Processing Systems 26 (NIPS 2013), Lake Tahoe, NV, USA, 5–10 December 2013.
14. Nguyen, T.H.; Grishman, R. Event Detection and Domain Adaptation with 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 2: Short Papers), Beijing, China, 26–31 July 2015; pp. 365–371.
15. Chen, Y.; Xu, L.; Liu, K.; Zeng, D.; Zhao, J. 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), Beijing, China, 26–31 July 2015; pp. 167–176.
16. Nguyen, T.H.; Grishman, R. Modeling Skip-Grams for Event Detection with Convolutional Neural Networks. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 1–5 November 2016; pp. 886–891.
17. Ghaeini, R.; Fern, X.; Huang, L.; Tadepalli, P. Event Nugget Detection with Forward-Backward Recurrent Neural Networks. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Berlin, Germany, 7–12 August 2016; pp. 369–373.
18. 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.
19. Liu, X.; Luo, Z.; Huang, H. Jointly Multiple Events Extraction via Attention-based Graph Information Aggregation. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 1247–1256.
20. Yan, H.; Jin, X.; Meng, X.; Guo, J.; Cheng, X. Event Detection with Multi-Order Graph Convolution and Aggregated Attention. 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), Hong Kong, China, 3–7 November 2019; pp. 5766–5770.
21. Cui, S.; Yu, B.; Liu, T.; Zhang, Z.; Wang, X.; Shi, J. Edge-Enhanced Graph Convolution Networks for Event Detection with Syntactic Relation. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2020, Online, 16–20 November 2020; pp. 2329–2339.
22. Lai, V.D.; Nguyen, T.N.; Nguyen, T.H. Event Detection: Gate Diversity and Syntactic Importance Scores for Graph Convolution Neural Networks. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020; pp. 5405–5411.
23. Rajpurkar, P.; Jia, R.; Liang, P. Know What You Don't Know: Unanswerable Questions for SQuAD. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Melbourne, Australia, 15–20 July 2018; pp. 784–789.
24. Dwork, C.; Feldman, V.; Hardt, M.; Pitassi, T.; Reingold, O.; Roth, A. The reusable holdout: Preserving validity in adaptive data analysis. *Science* **2015**, *349*, 636–638. [CrossRef] [PubMed]
25. Papernot, N.; McDaniel, P.; Swami, A.; Harang, R. Crafting adversarial input sequences for recurrent neural networks. In Proceedings of the MILCOM 2016-2016 IEEE Military Communications Conference, Baltimore, MD, USA, 1–3 November 2016; pp. 49–54.
26. Alzantot, M.; Sharma, Y.; Elgohary, A.; Ho, B.J.; Srivastava, M.; Chang, K.W. Generating Natural Language Adversarial Examples. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 2890–2896.
27. Ren, S.; Deng, Y.; He, K.; Che, W. Generating Natural Language Adversarial Examples through Probability Weighted Word Saliency. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 1085–1097.
28. Morris, J.; Yoo, J.Y.; Qi, Y. TextAttack: Lessons learned in designing Python frameworks for NLP. In Proceedings of the Second Workshop for NLP Open Source Software (NLP-OSS), Online, 19 November 2020; pp. 126–131.
29. Zeng, G.; Qi, F.; Zhou, Q.; Zhang, T.; Ma, Z.; Hou, B.; Zang, Y.; Liu, Z.; Sun, M. OpenAttack: An Open-source Textual Adversarial Attack Toolkit. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations, Online, 1–6 August 2021; pp. 363–371.

30. Wang, X.; Liu, Q.; Gui, T.; Zhang, Q.; Zou, Y.; Zhou, X.; Ye, J.; Zhang, Y.; Zheng, R.; Pang, Z.; et al. TextFlint: Unified Multilingual Robustness Evaluation Toolkit for Natural Language Processing. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations, Online, 1–6 August 2021; pp. 347–355.
31. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]
32. Lu, Y.; Lin, H.; Han, X.; Sun, L. Distilling Discrimination and Generalization Knowledge for Event Detection via Delta-Representation Learning. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August July 2019; pp. 4366–4376.
33. Liu, J.; Chen, Y.; Liu, K.; Jia, Y.; Sheng, Z. How Does Context Matter? On the Robustness of Event Detection with Context-Selective Mask Generalization. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2020, Online, 16–20 November 2020; pp. 2523–2532.
34. Miller, G.A. WordNet: A lexical database for English. *Commun. ACM* **1995**, *38*, 39–41. [CrossRef]
35. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language models are unsupervised multitask learners. *OpenAI Blog* **2019**, *1*, 9.
36. Zhu, H.; Xiao, K.; Ou, L.; Wang, M.; Liu, L.; Huang, H. Attention-Based Graph Convolution Networks for Event Detection. In Proceedings of the 2021 7th International Conference on Big Data and Information Analytics (BigDIA), Chongqing, China, 29–31 October 2021; pp. 185–190.
37. Manning, C.; Surdeanu, M.; Bauer, J.; Finkel, J.; 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.
38. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. In Proceedings of the 3rd International Conference on Learning Representations(ICLR), San Diego, CA, USA, 7–9 May 2015.