information

MDPI

Article

Semantic Enhanced Distantly Supervised Relation Extraction via Graph Attention Network

Xiaoye Ouyang [1,2], Shudong Chen [1,2,*] and Rong Wang [3]

[1] Institute of Microelectronics of the Chinese Academy of Sciences, Beijing 100049, China; ouyangxiaoye@ime.ac.cn
[2] University of Chinese Academy of Sciences, Beijing 100049, China
[3] Key Laboratory of Space Object Measurement Department, Beijing Institute of Tracking and Telecommunications Technology, Beijing 100049, China; zhr2019@ahou.edu.cn
* Correspondence: chenshudong@ime.ac.cn; Tel.: +86-010-82995546

Received: 11 October 2020; Accepted: 12 November 2020; Published: 14 November 2020

Abstract: Distantly Supervised relation extraction methods can automatically extract the relation between entity pairs, which are essential for the construction of a knowledge graph. However, the automatically constructed datasets comprise amounts of low-quality sentences and noisy words, and the current Distantly Supervised methods ignore these noisy data, resulting in unacceptable accuracy. To mitigate this problem, we present a novel Distantly Supervised approach SEGRE (Semantic Enhanced Graph attention networks Relation Extraction) for improved relation extraction. Our model first uses word position and entity type information to provide abundant local features and background knowledge. Then it builds the dependency trees to remove noisy words that are irrelevant to relations and employs Graph Attention Networks (GATs) to encode syntactic information, which also captures the important semantic features of relational words in each instance. Furthermore, to make our model more robust against noisy words, the intra-bag attention module is used to weight the bag representation and mitigate noise in the bag. Through extensive experiments on Riedel New York Times (NYT) and Google IISc Distantly Supervised (GIDS) datasets, we demonstrate SEGRE's effectiveness.

Keywords: distantly supervised; relation extraction; graph attention network; dependency tree

1. Introduction

Relation extraction aims to extract relations between pairs of marked entities in texts, which is one of the fundamental tasks in natural language processing (NLP) [1–3]. One primary problem of traditional supervised relation extraction (RE) methods is the requirement of large-scale manual labeling, which is very time-consuming and labor-intensive. Thus, Mintz et al. [4] proposed a Distantly Supervised RE, which constructs the dataset by aligning a known knowledge base (KB) and sentences crawled from web pages of the New York Times (NYT) automatically. Under the assumption that, if there is a relation between two entities in KB, then all sentences containing these two entities also represent the same relation. The problem of incorrect labeling often occurs. Riedel et al. [5] proposed a multi-instance learning method to relax this assumption. Despite the problem of wrong labeling, the Distantly Supervised methods still suffer from low-quality sentences, which are automatically generated by crawling web pages [6]. To handle the problem of low-quality sentences, we have to face two major challenges: (1) increase valuable auxiliary information; (2) reduce the noise of irrelevant words in the sentence. Noisy words in the text refer to words that do not contain semantics or words that have nothing to do with the information conveyed by the text. Non-noisy words refer to words

that contain semantics and are also a part of text semantics. When the noisy words are removed from the text, the rest are non-noisy words.

For the use of valuable auxiliary information, an idea model should make full use of local features or external information to extract precise semantic features from low-quality sentences containing noisy words. On the one hand, by encoding the position information of the word in the sentence, the position features of the corpus can be obtained. On the other hand, entity type information provides abundant background knowledge, which can be used to enhance semantics and the effectiveness of RE. For instance, the sentence "$[SeamlessWeb]_{e1}$ is a symbol of the heavy time commitment demanded by many of $[NewYork]_{e2}$'s professional service firms", the entity pair *SeamlessWeb* and *NewYork* has a relation */business/company/place_founded*, which is difficult to extract if the information *Seamlessweb* is a company and *new_york* is a location is lacking. Therefore, entity features learned from entity types are prior knowledge to initialize the RE model. We will use the entity type information to obtain more entity semantic features in this paper.

As for the other challenge, syntactic structure information is used to capture the semantic relationship and reduce the noise in the sentence. Figure 1 illustrates two methods for acquiring sentence semantics. In (a), the shortest dependent path between the entity pairs is displayed in the dependency tree and highlighted in bold (edges and marks). The dependency tree is used to express the dependency relationship between words in a sentence. Specifically, it analyzes and recognizes the grammatical components such as "subject-predicate-object" and "fixed adverbial complement" in the sentence. The dotted line is not the shortest path, but it also has a semantic impact on the critical path. Through the syntactic dependency graph, we can more intuitively discover the syntactic relationship between two entity pairs, reduce the interference of irrelevant words, which helps understand sentences and achieve more accurate relationship extraction. In (b), sequence structure refers to reading the words in a sentence sequentially from left to right. The sequence method is using adjacent words to obtain sentence semantics, which cannot obtain the direct connection between the keywords in the sentence. Comparing the advantages and disadvantages of the two methods, this paper chooses the dependency tree method, making full use of the syntactic structure to effectively analyze the semantic connection between the entity pairs, and judging the relationship between the entity pairs more reasonably.

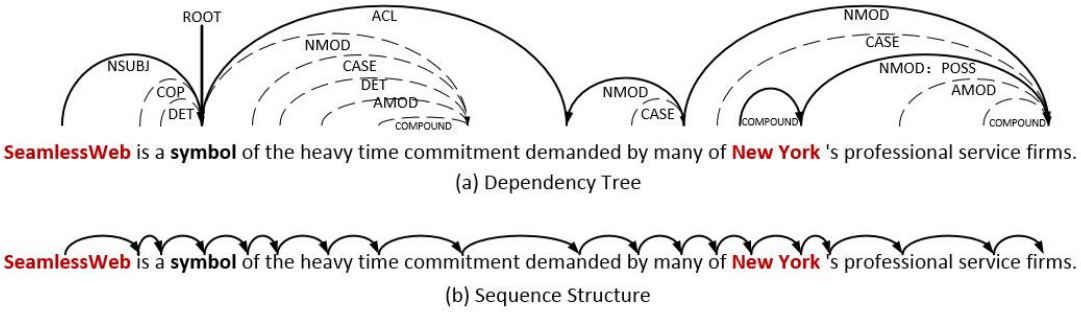

Figure 1. The example uses dependency tree and sequence structure to obtain sentence semantic, and assist in extracting relations between entities (indicated in red). In (**a**), the dependency tree can clearly express the dependency relationship between words in the sentence. Specifically, it analyzes and recognizes the grammatical components such as "subject-predicate-object" and "fixed adverbial complement" in the sentence. Each node is representing a word. In (**b**), the words in the sentence are read sequentially, usually from the left to the right, such as LSTM and GRU, while there are also two-way sequential reading forms, such as BiLSTM and BiGRU.

In this paper, we propose a novel semantic enhanced Distantly Supervised relation extraction method SEGRE, which utilizes additional semantic information and dependent syntactic to improve effective semantics against noisy words and reduce inner-sentence noise. For improving effective

semantics, SEGRE adopts word position and entity type information to provide abundant local features and background knowledge. Furthermore, it uses encoded syntactic information obtained from Graph Attention Networks along with embedded additional semantic information to improve neural relation extraction. Our contributions can be summarized as follows:

- We propose SEGRE, a novel semantic enhanced method for improving Distantly Supervised RE, which utilizes additional semantic features and knowledge learned from word position and entity type information to strengthen its robustness against low-quality corpus.
- To handle the problem of low-quality sentences, SEGRE uses Graph Attention Networks for modeling syntactic information and enhancing semantic features of important words, which has been shown to perform competitively.
- Experimental results show that SEGRE has achieved significant results on benchmark datasets, which improves the Precision/Recall (PR) curve area from 0.39 to 0.41 and increases P@100 by 4.7% over the state-of-the-art work.

The rest of this paper is organized as follows. Section 2 summarizes previous studies on relation extraction. Section 3 details the proposed model SEGRE and describes its various modules. Section 4 presents the experimental results. Section 5 concludes the paper.

2. Related Work

Relation extraction is a key component of constructing a relational knowledge graph and can be applied to structured search, sentiment analysis, question answering, and summary. Distantly Supervised relation extraction is proposed by Mintz et al. [4] to solve the problem of the lack of labeled training data. However, the sentence that refers to these two entities does not necessarily represent the relationship in the known knowledge base. Distantly Supervised inevitably causes an incorrect labeling problem. Thus, multi-instance learning methods are adopted to address this issue [5,7,8].

Deep neural network models are often used to perform relation extraction tasks. Here we introduce several basic types of deep neural networks: RNN [9] is widely used in the processing of time series data but has the problem of Long-Term Dependencies. The LSTM [10] model was born and used to improve this situation. biLSTM is a combination of the forward lstm and the backward lstm, which can encode front-to-back and back-to-front information and capture bidirectional semantics. LSTM also has many variants, among which the most used is GRU [11], which combines the forget gate and the input gate into a single update gate. The biGRU is a combination of the forward GRU and the backward GRU.

The large-scale automatically constructed dataset by crawling web pages will lead to the amount of low-quality sentences [12]. The use of additional semantic information provides abundant features and knowledge to enhance semantics against low-quality corpus. Zeng et al. [13] adopted piecewise convolution neural networks (PCNNs), which use the position information of words in a sentence to model the sentence representation. Yaghoobzadeh et al. [14] also tried to mitigate the noise in DS by combining entity type and relation extraction models. Vashishth et al. [15] used entity type and relation alias information to impose soft constraints when predicting relations. However, the above methods ignore inner-sentence noise.

As neural networks have been widely used, an increasing number of researches have been proposed. Lin et al. [16] proposed selective attention to neural network examples. Ji et al. [12] assigned more precise attention weights using entity descriptions. Nagarajan et al. [17] used attention to learn from multiple valid sentences. We also used attention mechanisms [18] to learn sentence and bag representations.

Moreover, features based on dependency trees are beneficial for relation extraction [4]. Xu et al. [19] adjusted the neural model to encode the shortest dependent path. Zhang et al. [20] adopted a path-centric pruning strategy. He et al. [21] established Subtree Parsing (STP) to delete noisy words that are not related to relations. Graph convolution network (GCN) [22] incorporates

structural information based on dependencies into the neural models. Song et al. [23] used GCN to directly encode the complete dependency graph. Zhang et al. [3] proposed Attention Guided Graph Convolution Networks (AGGCNs), a soft pruning method that automatically selects useful substructures. More recently, Velickovic et al. [24] proposed graph attention networks (GATs), which uses the attention mechanism to weight neighborhood states. The combination of reducing inner-sentence noise and using additional semantic information can better improve the performance of relation extraction.

Recently, scholars adopt feature extraction and text analysis methods in specific application scenarios to improve performance. Ali et al. [25] proposed a big data analytics engine based on data mining techniques, ontologies, and BiLSTM to improve healthcare monitoring accuracy. Ali et al. [26] used ensemble deep learning and feature fusion approaches to predict heart disease. Kaplan et al. [27] applied feature extraction technology to diagnose bearing vibration signals. Ayvaz et al. [28] studied to diminish the deficiency in the strategic cost management and prediction of economic crises with deep learning methods. Distantly supervised relation extraction is also beneficial for the construction of a knowledge graph in a specific application domain.

3. SEGRE Model (Semantic Enhanced GATs Relation Extraction)

An overview of the proposed SEGRE for Distantly Supervised relation extraction is illustrated in Figure 2. SEGRE consists of three modules used to learn the representation of a given bag and feed it into the softmax classifier. Firstly, the input sentences concatenate word, position and entity type embedding to encode the local context of each word and get the multi-level word representation. Secondly, we construct a syntactic dependency tree for each word in the sentence through a Bi-GRU and input it into the graph attention network to get the syntactic sentence representation. Furthermore, a group of bags sharing the same relation label in the training set is aggregated using the intra-bag attention module to weight the bag representation. Finally, the bag representation is fed to a softmax classifier to get the relation of the entity pair in the sentence. Each module will be described in detail in subsequent sections.

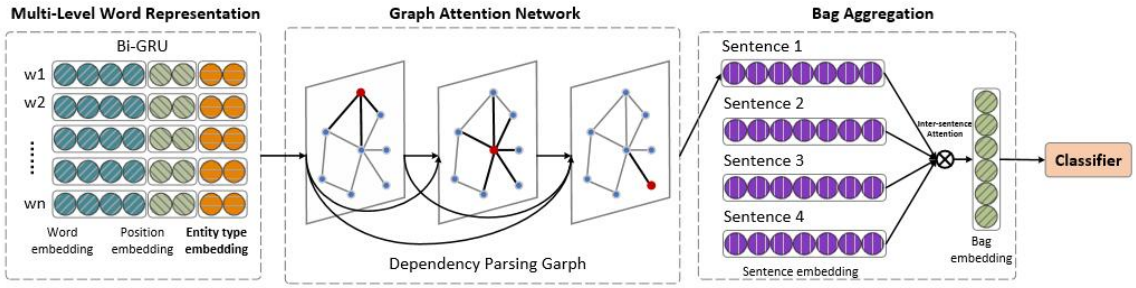

Figure 2. The framework of the proposed Semantic Enhanced Graph attention networks Relation Extraction (SEGRE). SEGRE first encodes each word in the sentence by concatenating word, position, and entity type information. Then the sentence representation is achieved by constructing a graph attention network using a syntactic dependency tree. Next, the bag representation is calculated by weighting sentence embeddings using intra-bag attention. Finally, the bag representation is fed to a softmax classifier to get the relation of the entity pair.

3.1. Multi-Level Word Representation

The multi-level word representation concatenates word information, position information, and entity type information. The word information is encoded by Bert [29] to obtain the semantics of the current word in the sentence. The position information records the position of the current word in the sentence, inspired by Zeng et al. [13]. When the word is in different positions, it represents different semantics and importance. Entity type information refers to the type to which the current word belongs. For example, the entity type of *Seamlessweb* is the company, so through the entity type

company you can know that *Seamlessweb* is the name of a company. Therefore, more meaningful word semantics can be obtained by using multi-level word representation. The specific implementation is as follows.

The inputs of the network are word, position tokens and entity type, which are transformed to the distributed representations before being input into the neural model. We extract meaningful word representations from different level semantics, i.e., the word embedding $e_w(w)$, the position embedding $e_p(w)$, and the entity type embedding $e_t(w)$.

For the word w in the sentence x, we represent each word by k dimensional Bert embedding.

In order to integrate the relative position of tokens with respect to target entities, we use p dimensional position embedding. Specifically, we use Pos1 and Pos2 to refer to the relative distance between the current word and the head and tail entities respectively. For instance, in Figure 1 relative distances of *symbol* from *seamlessweb* and *newyork* are 3 and -9 respectively. Then the position of each word is transformed to a *pdimensions*.

Entity types can enforce constraints on the prediction of the relation between subject and object. For instance, in Figure 1 the relation/business/company/place_founded can only exist between a *company* and a *location*. The entity type embedding refers to FIGER [30] by k_t dimensional embedding. Note that if the word in the sentence is not an entity, the entity type is completed with 0.

The final word presentation is obtained by concatenating these three parts of embeddings:

$$e(w) = [e_w(w) \oplus e_p(w) \oplus e_t(w)] \tag{1}$$

where $\oplus$ denotes the concatenation operation. Thus, we get a sequence of word vector $\{v_t\}$.

3.2. Bidirectional Gated Recurrent Unit

Based on the word vector $\{v_t\}$, we adopt a layer of bidirectional Gated Recurrent Unit (GRU) [11] to learn the semantic information of the sentence, which uses a hidden state vector $\{h_t\}$ to remember important signals. At each step, a new hidden state is computed based on previous hidden state using the same function.

$$z_t = \sigma(W_z V_t + U_z h_{t-1}) \tag{2}$$

$$r_t = \sigma(W_r V_t + U_r h_{t-1}) \tag{3}$$

$$\widetilde{h_t} = tanh(W_h V_t + U_h(r_t \Theta h_{t-1})) \tag{4}$$

$$h_t = (1 - z_t)\Theta h_{t-1} + z_t \Theta \widetilde{h_t} \tag{5}$$

where z_i and r_i are the update gate and reset gate, $\sigma(\cdot)$ is a sigmoid function, and W_z, W_r, W_h, U_z, U_r, U_h are parameters. $e(w_k|s)$ is the representation of w_k given s, which comes from the hidden vectors of $h_{k,t}$.

Furthermore, Bi-GRU that implements GRU in both forward and reverse can be used to access the long-distance semantics of the future and the past.

$$\overrightarrow{h_t} = GRU(w_k, \overrightarrow{h_{t-1}}) \tag{6}$$

$$\overleftarrow{h_t} = GRU(w_k, \overleftarrow{h_{t-1}}) \tag{7}$$

$$h^* = \alpha_t \overrightarrow{h_t} + \beta_t \overleftarrow{h_t} + b_t \tag{8}$$

where α_t and β_t represent the weights corresponding to the forward hidden layer state $\overrightarrow{h_t}$ and the reverse hidden state $\overleftarrow{h_t}$ at time t, and b_t indicates the hidden state bias at time t.

3.3. Graph Attention Network

Although Bi-GRU can capture local context, it fails to capture long-range dependencies that can be captured through dependency edges. We employ Graph Attention Networks for encoding features from syntactic dependency trees to improve relation extraction. The syntactic dependency tree is generated by Stanford CoreNLP [31].

We use the constructed syntactic dependency tree to form a graph, $\zeta(\nu, \varepsilon)$, where the nodes ν are the words in the sentence and the edges ε are the syntactic relations in the dependency tree. An edge from node u to node v with label l_{uv} is represented as $(\nu, \varepsilon, l_{uv})$. If there is a relation label existing between two words in the sentence, then the two words in the dependency graph are directly connected. Since the dependency tree has 55 different relation labels, which makes the constructed dependency graph too complicated. We use the same processing method as Nguyen and Grishman [32] to construct the graph, and only three kinds of edge labels are used to represent the relation, which are forward ($\rightarrow$), backward ($\leftarrow$), self-loop ($\perp$), defined as follows:

$$l_{uv} = \begin{cases} \rightarrow & \text{if edge is a forward edge} \\ \leftarrow & \text{if edge is a backward edge} \\ \perp & \text{if edge is a selfloop edge} \end{cases} \tag{9}$$

The input of GATs is $h^* = \{h_1^*, h_2^*, ..., h_m^*\}$, where m is the number of words in sentence. e_{ij} represents the importance of the characteristics of node j to node i. We put up the structure of the dependency graph and only calculate the e_{ij} where node j is adjacent to node i in the graph. In order to make coefficients easy to compare between different nodes, we use the softmax function to normalize them across all choices of j. For each word w_i, GATs embedding h_i^{gat} is defined as:

$$\alpha_{ij} = softmax_j(e_{ij}) = \frac{exp(LeakyReLU(a^{-\tau}[Wh_i^* || Wh_j^*]))}{\sum_{k \in N_i} exp(LeakyReLU(a^{-\tau}[Wh_i^* || Wh_j^*]} \tag{10}$$

where the single-head attention mechanism α_{ij} is a single-layer feedforward neural network and applies the LeakyReLU nonlinearity [24]. LeakyReLU activation function is a variant of the ReLu activation function, and ReLu is the most commonly used activation function in neural networks. The LeakyReLU activation function has a small slope for negative inputs, and because the derivative is always non-zero, it can reduce the appearance of silent neurons and allow gradient-based learning, and solves the problem that neurons do not learn after the Relu function enters the negative interval. α_{ij}^k are normalized attention coefficients computed by the kth attention mechanism α^k, and W^k is the corresponding input linear transformation's weight matrix. The syntactic graph encoding from GATs and Bi-GRU output vector are concentrated to obtain the final sentence representation $h_i^{concat} = [h_i^*; h_i^{gat}]$.

3.4. Bag Aggregation

In this section, the first step of bag aggregation is to calculate the weight of different sentences in the bag through the intra-bag attention mechanism, and the second step is to multiply the sentence embedding and its weight and then accumulate to get the bag representation. After bag aggregation, the bag representation is sent to the softmax classifier to obtain the classification of the relationship between entities.

For utilizing all valid sentences, we employ the attention mechanism used by Jat et al. [33] over sentences to obtain a representation for the entire bag. For sentence s_i in the bag, attention weight α_i is calculated as follows:

$$\alpha_i = \frac{exp(s_i)}{\sum_{j=1}^{n} exp(s_j)} \tag{11}$$

Bag representations B are calculated by weighting sentence embedding using intra-bag attention, which can deal with noise at sentence-level.

$$B = \sum_{j=1}^{m} \alpha_i h_i^{concat} \tag{12}$$

Finally, the bag representation is fed to the softmax classifier to obtain the probability distribution of different relations.

$$p(y) = Softmax(W \cdot B + b) \tag{13}$$

4. Experiments

In order to demonstrate the performance and adaptability of SEGRE, we compare several methods on two benchmark datasets, Riedel New York Times (NYT) and Google IISc Distantly Supervised (GIDS) datasets, and give implementation details and experimental results analysis.

4.1. Compared Methods

We have chosen seven methods to compare their performance with the proposed SEGRE. Mintz [4] first proposes a multi-class logistic regression model for Distantly Supervised; MultiR [7] uses a probabilistic graphical strategy for multi-instance learning; MIMLRE [8] jointly models multiple instances and multiple tags. PCNN [13] adopts a relation extraction model combining piecewise and CNN. PCNN + ATT [16] uses PCNN and attention mechanisms to obtain sentence representations. BGWA [33] adopts a word and sentence level attention strategy for relation extraction. RESIDE [15] applies entity type and relation alias information to impose soft constraints.

In addition, we also change the partial structure of SEGRE, and compare the performance of three variations of the proposed SEGRE. Specifically, $SEGRE_{GAT*}$ uses undirected edges to construct GAT instead of directed edges; $SEGRE_{GCN}$ uses GCN to embed sentence dependency information instead of GAT; $SEGRE_{type^-}$ removes the entity type information in multi-level word representation; and $SEGRE_{att^-}$ implements bag representation without an attention mechanism.

4.2. Data Sets

We evaluated SEGRE on Riedel NYT [5] and GIDS datasets. Riedel NYT dataset has been widely used for RE by keeping the relation between Freebase and the New York Times Corpus consistent, using sentences in 2005–2006 to create training sets and sentences in 2007 for test sets. The entities were annotated with the Stanford NER tool [34] and linked to Freebase.

The GIDS dataset was created by Jat et al. [33], which extends the Google relation extraction corpus with other instances of each entity pair. The GIDS dataset guarantees the "at-least-one" assumption of multi-instance learning, which makes automatic evaluation more reliable, thereby eliminating the need for manual verification. The corpora statistics of the two datasets are shown in Table 1. There are 53 types of relations between entity pairs in the Riedel NYT dataset, and 5 types of GIDS datasets. The training set (TRAIN), validation set (VALID) and testing set (TEST) are officially segmented.

Table 1. Corpora Statistics for the Riedel New York Times (NYT) and Google IISc Distantly Supervised (GIDS) datasets.

Datasets		TRAIN	DEV	TEST
Riedel NYT	sentences	455,771	114,317	172,448
	entities	233,064	58,635	96,678
GIDS	sentences	11,297	1864	5663
	entities	6498	1082	3247

4.3. Implementation Details

If the comparison methods and SEGRE are implemented in an identical experimental environment, we directly copy the results of these experiments, otherwise the methods will be reproduced in the context of this paper.

SEGRE uses TensorFlow libraries and python 3. We used cross-validation to tune our model and grid search for super-parameter optimization, and chose the best performance setting as the final setting. In this experiment, we applied the Adam optimizer with the learning rate decay. GRU size $m = 230$, position embedding size $p = 16$, entity type embedding size $k = 50$. To avoid hyperparameters, we adopted the 38 coarse-grained types of FIGER's first layer instead of all 112 fine-grained entity types.

4.4. Experimental Results

In order to evaluate the effectiveness of our proposed SEGRE, we compared it with the method described in Section 4.1. We use the Precision–Recall curve and top-N precision (P@N) metric to evaluate the performance in our experiments. Notice that we only use the neural compared methods on the GDS dataset.

The Precision–Recall curves on Riedel NYT and GIDS are shown in Figure 3. We found that SEGRE achieved higher accuracy in the entire recall range of both datasets. On the Riedel NYT dataset, all non-neural network methods are not very effective, because they use existing NLP tools for feature extraction, which may produce errors. The PR curve areas of PCNN, PCNN + ATT, BGWA, and RESIDE are about 0.332, 0.386, 0.394 and 0.409 respectively, while SEGRE increases it to 0.417. Meanwhile, on the GIDS dataset, the PR curve areas of PCNN, PCNN + ATT, BGWA, and RESIDE are about 0.694, 0.743, 0.751, and 0.787 respectively, while SEGRE increases it to 0.791. The result indicates that our SEGRE can use word position and entity type information to increase additional semantic information, and use syntactic dependency trees to eliminate unrelated noise words in sentences, it finally achieves more accurate sentence representations for relationship extraction.

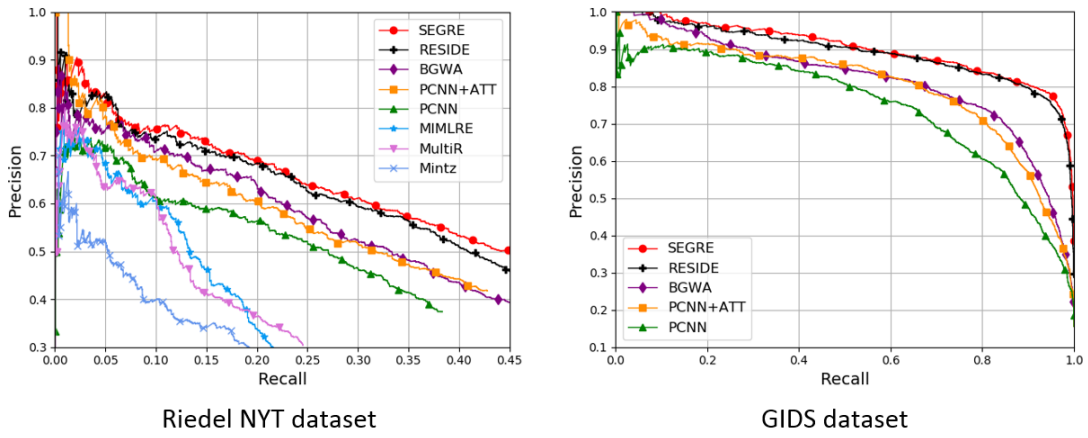

Figure 3. Comparison of Precision–Recall curves. SEGRE achieves higher precision over the entire range of recall than all the baselines on both datasets.

Following previous works, we adopt P@N as a quantitative indicator to compare our model with baselines based on various instances under each relational tuple. P@N means the precision of the relation classification results with the top N highest probabilities in the test set. Table 2 shows the P@N value of relation extraction as the number of sentences in the bag changes. Here, one, two and all represent the number of sentences randomly selected from the package, forming three types of data sets. The table shows the P@100, P@200, P@300, and their means of the SEGRE model and its compared methods on the test sets. We can see our proposed methods achieved higher P@N values

than previous work, and the P@100, P@200 and P@300 values of SEGRE have been improved to 3.6%, 2.9%, 1.7% over state-of-the-art model, respectively.

Table 2. P@N for relation extraction using a variable number of sentences in bags in the Riedel dataset.

	One				Two				All			
	P@100	**P@200**	**P@300**	**Mean**	**P@100**	**P@200**	**P@300**	**Mean**	**P@100**	**P@200**	**P@300**	**Mean**
PCNN	73.3	64.8	56.8	65.0	70.3	67.2	63.1	66.9	72.3	69.7	64.1	68.7
PCNN + ATT	73.3	69.2	60.8	67.8	77.2	71.6	66.1	71.6	76.2	73.1	67.4	72.2
BGWA	78.0	71.0	63.6	70.9	81.0	73.0	64.0	72.7	82.0	75.0	72.0	76.3
RESIDE	80.0	75.5	69.3	74.9	83.0	73.5	70.6	75.7	84.0	78.5	75.6	79.4
SEGRE	82.6	74.3	68.3	75.1	84.9	78.7	73.5	79.0	87.6	81.4	77.3	82.1

Figure 4 shows the performance of different ablated versions of our proposed SEGRE on the Riedel NYT and GIDS datasets. We observe that after SEGRE changes different components, the performance of the model varies significantly. The PR curve area of SEGRE is 0.007 higher than that of $SEGRE_{GAT^*}$ on the NYT dataset, and 0.023 higher than that of $SEGRE_{GAT^*}$ on the GIDS dataset. Because the syntactic dependency tree constructed in this paper is directional. The direction information includes the relationship between words in the text. The directed edge in GAT can better reflect the syntactic dependency tree structure than the undirected edge. The PR curve area of SEGRE is 0.015 higher than that of $SEGRE_{GCN}$ on the NYT dataset, and 0.046 higher than that of $SEGRE_{GCN}$ on the GIDS dataset. This result confirms that GAT effectively encodes grammatical information and removes irrelevant word noise in sentences. In addition, the PR curve area of SEGRE is 0.053 higher than that of $SEGRE_{type^-}$ on the NYT dataset, and 0.128 higher than that of $SEGRE_{type^-}$ on the GIDS dataset. The introduction of entity type information indicates that it supplements text features and can enhance the relationship extraction performance. Further, the PR curve area of SEGRE is 0.031 higher than $SEGRE_{att^-}$ on the NYT dataset, and 0.057 higher than $SEGRE_{att^-}$ on the GIDS dataset. This proves that the attention mechanism in the bag helps reduce the noise between sentences. In conclusion, the entity type information has the greatest impact on the performance of the model because it provides additional semantics and is very helpful for the task.

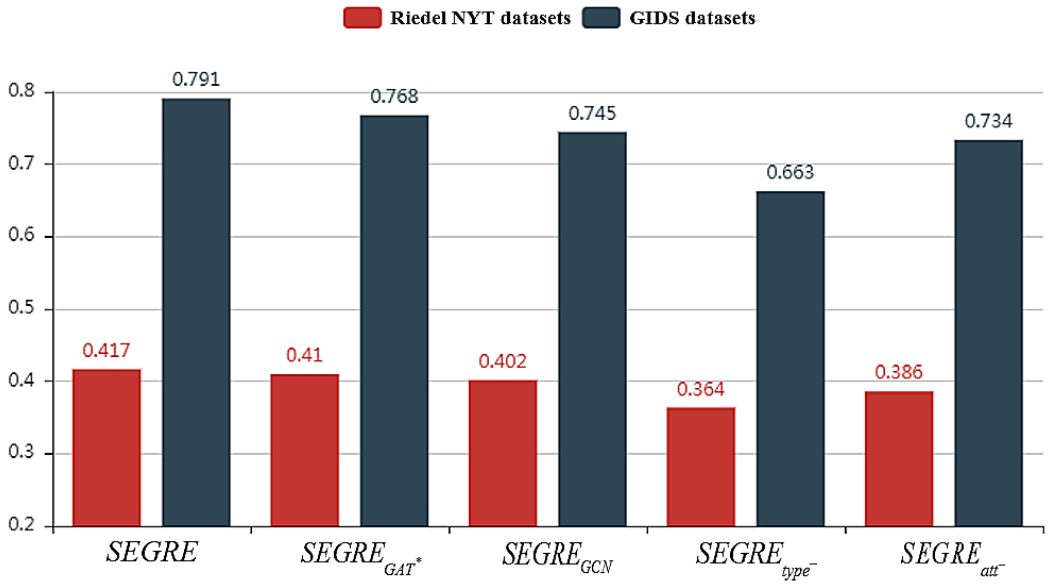

Figure 4. Performance comparison of different SEGRE ablated version on two datasets.

5. Conclusions

In this paper, we propose SEGRE, a novel semantic enhanced approach for Distantly Supervised relation extraction. It aims at dealing with the low-quality datasets by increasing valuable additional

semantic information and reducing the noise of irrelevant words in the sentence. Compared with other methods, the main innovations of the proposed method are as follows: In the word representation stage, SEGRE uses multi-level word representation, including word information, position information, and entity type information, which enriches the semantics contained in a word embedding. In the sentence representation stage, SEGRE uses a graph attention network, which extracts important information more effectively than a graph convolutional network and reduces noise in a sentence. In the bag representation stage, SEGRE added an intra-bag attention mechanism to calculate the representation of the bag, reducing the noise in the bag. SEGRE increases valuable semantic information throughout all stages of the model. Experimental results show that SEGRE achieves state-of-the-art results on two benchmark datasets.

Using graph neural networks to extract sentence semantics is our preliminary study on Relation extraction. We only considered semantic analysis at the sentence level, but future work should focus on the document level. Furthermore, the information contained in the document will be richer than a single sentence, but it will also bring more noise. Future work should reduce document-level noise and improve the effective use of document-level information.

Author Contributions: Conceptualization, X.O. and S.C.; data curation, X.O. and R.W.; formal analysis, X.O.; investigation, X.O.; methodology, X.O. and S.C.; project administration, S.C.; resources, S.C. and R.W.; software, X.O.; supervision, S.C.; validation, Xiaoye Ouyang, S.C. and R.W.; visualization, X.O.; writing—original draft, X.O.; writing—review and editing, X.O., S.C. and R.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research supported by the National Nature Science Foundation of China (No.61876144) and project (XDC02070600) from the Chinese Academy of Sciences.

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

Abbreviations

The following abbreviations are used in this manuscript:

SEGRE	Semantic Enhanced Graph attention networks Relation Extraction
GATs	Graph Attention Networks
NYT	New York Times
GIDS	Google IISc Distantly Supervised
RE	Relation extraction
NLP	Natural Language Processing
KB	Knowledge Base
RNN	Recurrent neural network
LSTM	Long short-term memory
BiLSTM	Bidirectional long short-term memory
GCN	Graph Convolution Network
GRU	Gated Recurrent Unit
biGRU	Bidirectional Gated Recurrent Unit
PCNNs	piecewise convolutional neural networks
STP	Subtree Parsing
AGGCNs	Attention Guided Graph Convolution Networks
ReLU	Rectified linear unit
P@N	top-N precision

References

1. Miwa, M.; Bansal, M. End-to-end relation extraction using LSTMs on sequences and tree structures. In Proceedings of the Meeting of the Association for Computational Linguistics, Berlin, Germany, 7–12 August 2016; Volume 1, pp. 1105–1116.
2. Verga, P.; Strubell, E.; Mccallum, A. Simultaneously self-attention to all mentions for full-abstract biological relation extraction. *arXiv* **2018**, arXiv:1802.10569.

3. Zhang, Y.; Guo, Z.; Lu, W. Attention guided graph convolutional networks for relation extraction. *arXiv* **2019**, arXiv:1906.07510.
4. Mintz, M.; Bills, S.; Snow, R.; Jurafsky, D. Distant supervision for relation extraction without labeled data. In Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, Suntec, Singapore, 2–7 August 2009; pp. 1003–1011.
5. Riedel, S.; Yao, L.; Mccallum, A. Modeling relations and their mentions without labeled text. In Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Barcelona, Spain, 19–23 September 2010; pp. 148–163.
6. Yang, W.; Ruan, N.; Gao, W.; Wang, K.; Ran, W.S.; Jia, W.J. Crowdsourced time-sync video tagging using semantic association graph. In Proceedings of the 2017 IEEE International Conference on Multimedia and Expo (ICME), Hong Kong, China, 10–14 July 2017; pp. 547–552.
7. Hoffmann, R.; Zhang, C.; Ling, X.; Zettlemoyer, L.; Weld, D. Knowledge based weak supervision for information extraction of overlapping relations. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, Portland, OR, USA, 19–24 June 2011; pp. 541–550.
8. Surdeanu, M.; Tibshirani, J.; Nallapati, R.; Manning, C.D. Multi-instance multi-label learning for relation extraction. In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing, Stroudsburg, PA, USA, 12–14 July 2012; pp. 455–465.
9. Zaremba, W.; Sutskever, I.; Vinyals, O. Recurrent neural network regularization. *arXiv* **2014**, arXiv:1409.2329.
10. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
11. Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv* **2014**, arXiv:1406.1078.
12. Ji, G.; Liu, K.; He, S.; Zhao, J. Distant supervision for relation extraction with sentence-Level attention and entity descriptions. In Proceedings of the National Conference on Artificial Intelligence, Melbourne, Australia, 19–25 August 2017; pp. 3060–3066.
13. Zeng, D.; Liu, K.; Chen, Y.; Zhao, J. Distant supervision for relation extraction via piecewise convolutional neural networks. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015; pp. 1753–1762.
14. Yaghoobzadeh, Y.; Adel, H.; Schutze, H. Noise mitigation for neural entity typing and relation extraction. *arXiv* **2016**, arXiv:1612.07495.
15. Vashishth, S.; Joshi, R.; Prayaga, S.; Bhattacharyya, C.; Talukdar, P. RESIDE: Improving distantly-supervised neural relation extraction using side information. *arXiv* **2018**, arXiv:1812.04361.
16. Lin, Y.; Shen, S.; Liu, Z.; Luan, H.; Sun, M. Neural relation extraction with selective attention over instances. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Berlin, Germany, 7–12 August 2016; Volume 1, pp. 2124–2133.
17. Nagarajan, T.; Jat, S.; Talukdar, P. CANDiS: Coupled attention-driven neural distant supervision. *arXiv* **2017**, arXiv:1710.09942.
18. Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv* **2014**, arXiv:1409.0473.
19. Xu, K.; Feng, Y.; Huang, S.; Zhao, D. Semantic relation classification via convolutional neural networks with simple negative sampling. *arXiv* **2015**, arXiv:1506.07650.
20. Zhang, Y.; Qi, P.; Manning, C. Graph convolution over pruned Dependency trees improves relation extraction. *arXiv* **2018**, arXiv:1809.10185.
21. He, Z.; Chen, W.; Li, Z.; Zhang, M.; Zhang, W.; Zhang, M. SEE: Syntax-aware entity embedding for neural relation extraction. *arXiv* **2018**, arXiv:1801.03603.
22. Defferrard, M.; Bresson, X.; Vandergheynst, P. Convolutional neural networks on graphs with fast localized spectral filtering. *arXiv* **2016**, arXiv:1606.09375.
23. Song, L.; Zhang, Y.; Wang, Z.; Gildea, D. N-ary relation extraction using graph state LSTM. *arXiv* **2018**, arXiv:1808.09101.
24. Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; Bengio, Y. Graph attention networks. *arXiv* **2017**, arXiv:1710.10903.

25. Ali, F.; El-Sappagh, S.H.A.; Islam, S.R.; Ali, A.; Attique, M.; Imran, M.; Kwak, K.-S. An intelligent healthcare monitoring framework using wearable sensors and social networking data. *Future Gener. Comput. Syst.* **2020**, *114*, 23–43. [CrossRef]
26. Ali, F.; El-Sappagh, S.H.A.; Islam, S.R.; Kwak, D.; Ali, A.; Imran, M.; Kwak, K.-S. A smart healthcare monitoring system for heart disease prediction based on ensemble deep learning and feature fusion. *Inf. Fusion* **2020**, *63*, 208–222. [CrossRef]
27. Kaplan, K.; Kaya, Y.; Kuncan, M. An improved feature extraction method using texture analysis with LBP for bearing fault diagnosis. *Appl. Soft Comput.* **2020,** *87*, 106019. [CrossRef]
28. Ayvaz, E.; Kaplan, K.; Kuncan, M. An Integrated LSTM Neural Networks Approach to Sustainable Balanced Scorecard-Based Early Warning System. *IEEE Access* **2020**, *8*, 37958–37966. [CrossRef]
29. Devlin, J.; Chang, M.; Lee, K.; Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
30. Ling, X.; Weld, D. Fine-grained entity recognition. In Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, Toronto, ON, Canada, 22–26 July 2012; pp. 94–100.
31. 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.
32. Nguyen, T.; Grishman, R. Graph convolutional networks with argument-aware pooling for event detection. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; pp. 5900–5907.
33. Jat, S.; Khandelwal, S.; Talukdar, P. Improving distantly supervised relation extraction using word and entity based attention. *arXiv* **2018**, arXiv:1804.06987.
34. Finkel, J.; Grenager, T.; Manning, C. Incorporating non-local information into information extraction systems by gibbs sampling. In Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05), Ann Arbor, MI, USA, 25–30 June 2005; pp. 363–370.