symmetry

MDPI

Article

Attention-Based LSTM with Filter Mechanism for Entity Relation Classification

Yanliang Jin [1,*], Dijia Wu [2] and Weisi Guo [3]

[1] Associate Professorship with the School of Communication and Information Engineering (SCIE), Shanghai University (SHU), Shanghai 200000, China
[2] School of Communication and Information Engineering (SCIE), Shanghai University (SHU), Shanghai 200000, China; bosco@shu.edu.cn
[3] School of Engineering, University of Warwick, Coventry CV4 7AL, UK; weisi.guo@warwick.ac.uk
* Correspondence: wuhaide@shu.edu.cn

Received: 17 September 2020; Accepted: 14 October 2020; Published: 19 October 2020

Abstract: Relation classification is an important research area in the field of natural language processing (NLP), which aims to recognize the relationship between two tagged entities in a sentence. The noise caused by irrelevant words and the word distance between the tagged entities may affect the relation classification accuracy. In this paper, we present a novel model multi-head attention long short term memory (LSTM) network with filter mechanism (MALNet) to extract the text features and classify the relation of two entities in a sentence. In particular, we combine LSTM with attention mechanism to obtain the shallow local information and introduce a filter layer based on attention mechanism to strength the available information. Besides, we design a semantic rule for marking the key word between the target words and construct a key word layer to extract its semantic information. We evaluated the performance of our model on SemEval-2010 Task8 dataset and KBP-37 dataset. We achieved an F1-score of 86.3% on SemEval-2010 Task8 dataset and F1-score of 61.4% on KBP-37 dataset, which shows that our method is superior to the previous state-of-the-art methods.

Keywords: relation classification; attention mechanism; bidirectional LSTM network; natural language processing

1. Introduction

Relation classification is an important natural language processing (NLP) task. It is the key step in many natural language applications such as information extraction [1,2], construction of knowledge base [3,4], and question answering [5,6]. Relation classification aims to extract valid information and classify the relationships between entities in sentences.

As shown in Figure 1, the sentence contains an example of the cause–effect (e1-e2) relation between the nominal "women" and "accident". <e1>,</e1>,<e2>,</e2> are four position indicators which specify the starting and ending of the nominals [7]. It is obvious that the relation is easy to extract when the sentence length is short. However, it may be difficult for classification when facing long sentences. We take one more sentence for example.

Sentence: "The <e1>woman</e1> that caused the <e2>accident</e2> was on the cell phone."		
Entity 1: woman	**Entity 2**: accident	**Relation**: Cause–Effect (e1-e2)

Figure 1. Example of a short sentence.

As shown in Figure 2, the sentence contains a relation of Entity–Origin (e1,e2) between the nominal "lawsuits" and "fans" and the distance of the entities is long, which increases the difficulty

of predicting a semantic relationship between two tagged entities. The larger the distance between entities, the longer the sentence length, and the greater the difficulty of relationship classification.

Sentence: "Lawyers in Detroit also worked overtime as several <e1>lawsuits</e1> ensued from angry and injured <e2>fans</e2>."		
Entity 1: lawsuits	**Entity 2**: fans	**Relation**: Entity–Origin(e1,e2)

Figure 2. Example of a long sentence.

In a sentence, in addition to the distance between the two target words introducing noise, the unrelated words between the two words also introduce noise. As shown in Figure 2, in this sentence, "lawsuits ensued from angry" will affect the classification results because it cannot be inferred directly for classification. From the grammatical level, the subject and the object are connected by the predicate, so the relationship can be judged indirectly by predicates. However, some adverbials, numerals, and pronouns may be involved to bring noise to the extraction of subject predicate object structure.

Machine learning used to be an effective way to solve relation classification problems. Traditional methods based on pattern recognition use artificial features or human-designed kernels to achieve high performance [8–10]. But there are two problems: (1) constructing features is time-consuming and (2) noise will be introduced in the process of constructing features. When we use external NLP tools like Wordnet to extract advanced features, such as shortest dependency path [11,12], named entity [13], part-of-speech tagging [14], etc., at the same time it will bring some noise information and affect generalization ability. Some recent work in the relation classification field focus on the use of deep neural networks to construct features, which is a more effective approach. There are two main neural network-based methods: recursive neural network (RNN) [15–17] and convolutional neural network (CNN) [18], which are used to train the end-to-end models. In addition, attention-based models also have a good performance in relation classification [19]. Apart from the methods mentioned above, some of other works are based on a combination of CNN and RNN to do relation classification tasks [20]. Commonly, they all use the deep learning method to classify the relation more effectively, reducing the time to construct artificial structural features [21,22] and improving the accuracy. Although these models achieved some improvement, several problems still existed. One problem is that the input noise may be amplified by the model and affect the results. Limited by the size of convolution kernel, CNNs can extract local structure but lose some effective information due to the size of receptive field. RNNs can make the network extract semantic information more accurately in sequence signal analysis. But at the same time, unnecessary feature information will also be introduced into the RNN network to affect the accuracy. In order to solve the problem of gradient vanishing [23] and for the method to work better with long sentences, attention mechanisms [24] and an RNN structure are proposed in the literature [23]. Another problem is that the proposed methods may only work for the specific dataset. Some experiments evaluated on only one dataset, which may have poor portability in other datasets. Especially when facing long sentences or complex nouns, the generalization ability is weak in predicting the relationship between two tagged entities. Yan et al. [25] proposed multichannel long short term memory (LSTM) networks, which allows for effective information integration from heterogeneous sources over the dependency paths.

In this paper, we propose a novel model, the multi-head attention long short term memory (LSTM) network with filter mechanism (MALNet), and present details of an experiment conducted on two different datasets. Our chief contributions are as follows:

1. We propose a novel two-channel neural network framework for the task of relation classification. One channel is to concatenate the word level feature based on attention mechanism. Another channel is to introduce entity context information for filtering noise, leaving the useful information for classification.

2. Our model differs from most previous models for relation classification, as they rely on the high-level lexical such as dependency parser, part-of speech (POS) tagger, and named entity recognizers

(NER) obtained by NLP tools like WordNet; we designed an interactive sentence level attention filter architecture to leave effective local feature information and designed a semantic rule to extract key word for learning more complicated features.

3. We conducted experiments using the SemEval-2010 Task 8 dataset and KBP-37 dataset. The experimental results demonstrate that our MALNet model performs better than the previous state-ot-the-art methods on both datasets. The remainder of this paper is structured as follows. Section 2 presents the related work about the relation classification. Section 3 introduces the architecture of our MALNet model. Section 4 provides the experiment setting and results. The conclusion is discussed in Section 5.

2. Related Work

Due to the practical significance of relation classification, a lot of research has been devoted to it. In recent years, deep neural networks have shown good performance on relation classification [8,21]. In building high-level features, it experienced a shift from human-designed construction to deep learning [9]. In the feature-based methods, such as named entities [13], shortest dependency path [10,11], and left or right tokens of the tagged entity [21], are applied to this field, but in the process of constructing the features it will cause accumulative errors and introduce noisy information.

For convolutional neural network (CNN) techniques, Zeng et al. [21] proposed a deep CNN to address this task. They used a CNN to model sentence-level features and lexical level features, including entities, left and right tokens of entities, and WordNet hypernyms of entities. Santos et al. [18] proposed a classification by ranking CNN (CR-CNN) model using a new rank loss to reduce the impact of artificial classes. It shows that new rank loss performs better than that of common cross-entropy loss function. Huang et al. [26] proposed an attention-based CNN (Attention-CNN) for semantic relation extraction, which employs a word-level attention mechanism to get the critical information for relation representation. These methods have limitations on learning sequence features because of the shortages of convolution kernels. In addition, convolutional neural networks are less effective in classifying long sentence relationships.

On the other hand, the RNN-based models show outstanding performance in processing text sequences in relation classification. Zhang et al. [16] proposed bidirectional long short-term memory networks (BLSTM) to capture the context information. Bidirectional LSTM has better performance than standard LSTM. Zhang et al. [27] proposed an RCNN (Recurrent Convolutional Neural Networks) model, which combines the advantages of RNN and CNN. It not only solved the problem of long-time dependence with RNN, but also extracted more abundant features with a CNN. Zhou et al. [15] used attention-based bidirectional long short-term memory networks to capture the most important semantic information in a sentence. This model does not rely on NLP tools to get lexical resources, which obtained state-of-the-art performance. The existing research indicates that the RNNs perform better than CNNs due to its context sensitivity. Recently, some researchers have proposed attention-based models [23]. Cao et al. [28] exploited a bidirectional long short-term memory network with adversarial training to extract sentence level features. In order to enhance the robustness, they leverage attention mechanisms to better learn the most influential features. Lee et al. [29] proposed a model bidirectional LSTM networks with entity-aware attention to learn more semantic features. Yan et al. [25] presented the shortest dependency path (SDP)-LSTM, a novel neural network to classify the relation of two entities in a sentence. The architecture leverages the shortest dependency path (SDP) between two entities, multichannel recurrent neural networks, with long short term memory (LSTM) units, to pick up heterogeneous information along the SDP.

We summarize the past methods and problems, and then put forward the model MALNet. This model can enhance the robustness and capture context information for relation classification task.

3. Our Model

In this section, we give an overview of the MALNet model. We introduce an attention-based BLSTM layer to extract word-level features and construct a sentence level attention filter layer to leave available information. We designed a semantic rule for the method of extracting key words. As shown in Figure 3, our model consists of five main components:

- Word Representation Layer: The sentence is mapped into a real-valued vector named word embeddings.
- Attention-based BLSTM Layer: This layer consists of two channels for extracting the world level features. One channel uses Bidirectional LSTM to capture context information and focus on the available features by attention mechanism Another channel directly utilizes attention mechanism to capture the similarity between words. We construct two channels to fit the feature better.
- Sentence Level Attention Filter Layer: This layer constructs a filter module to process the noise. We take all lexical level features into account aim to filter out noise and retain effective information for classification.
- Key Word Layer: In this layer, we analyze the sentence structure between the two target words, extract the key words for convolution, and provide auxiliary effects for classification.
- Classification Layer: High-level features will be fed into this layer. We calculate the relationship score to identify the relationship.

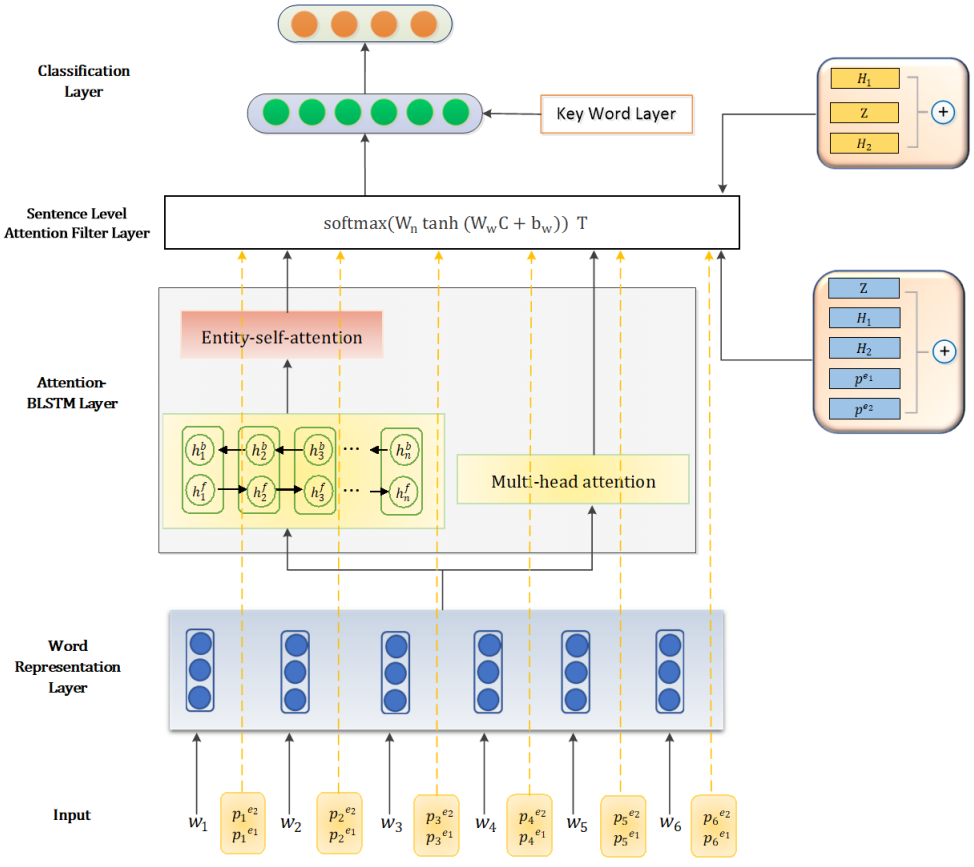

Figure 3. The whole architecture of our proposed multi-head attention long short term memory (LSTM) network with filter mechanism (MALNet) model.

(1) Word Representation Layer

Let us consider an input sentence denoted by $S = \{w_1, w_2, ..., w_n\}$, where n is the number of words. To represent a word, we embed each word into a low dimensional real-value vector called

word embeddings [30]. v_i is one-hot encoding of w_i, word embeddings can encode sparse one-hot representation v_i into real-valued vector by looking up in the matrix $W \in \mathbb{R}^{d^w * |V|}$, where d^w represents the dimension of the word vectors and V represents the word vocabulary size. The word representation $X = \{x_1, x_2, ..., x_n\}$ are mapped by the word w_i to a low dimensional real-value vector x_i, which are fed into the next layer. Our experiments directly utilize pre-trained weights of the publicly available embedding from language models (ELMO) [31]. Bidirectional language model can obtain the context representation of the current word and fine-tune word embeddings for relation classification.

(2) Attention-based BLSTM Layer

To capture the word-level features effectively, we design a two-channel module to extract it. In the first channel we use multi-head attention to extract word-level features. Since attention mechanism neglects the order of the sequence [23], we use it to capture the shallow features. Regardless of the length of sentences and the distance between entities, attention mechanism aims at modeling the strength of relevance between representation pairs [32].

We can regard an attention mechanism part as a mapping of a query and key-value pairs to an output. An attention function of the query and the key is adopted to compute the weight of each value, then the output is determined as a weighted sum of the values [23].

For multi-head attention, we use the word representation $X = \{x_1, x_2, ..., x_n\}$ to initialize query Q, key K and value V. Given a matrix of query Q, key K, and value V, the scaled dot-product attention is calculated by the following equation:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{(d_k)}}\right)V \tag{1}$$

$$\text{head}_\text{i} = \text{Attention}(W_i^Q Q, W_i^K K, W_i^V V) \tag{2}$$

$$Z = W^M[\text{head}_1 \oplus \cdots \oplus \text{head}_\text{r}] \tag{3}$$

where $W_i^Q, W_i^K, W_i^V \in \mathbb{R}^{d^k/r * d^k}$ are the projections matrices, $W^M \in \mathbb{R}^{d^k * d^k}$ is a mapping parameter from input space to representation space, r is the number of the attention heads, d_k is the dimension of the word vectors, the $\oplus$ represents the connection operator.

In the second channel, we combine the long short-term memory (LSTM) recurrent neural networks with the attention mechanism. Long short-term memory recurrent neural networks are an improvement over the general recurrent neural networks [24], which achieved good results and possessed a vanishing gradient problem. However, standard LSTM networks process monotonic sequences in time order, and can only capture information from left-to-right or from right-to-left; it splits the context information. So, we chose Bidirectional LSTM networks to capture the feature.

As shown in Figure 4, the BLSTM network contains two sub-networks for the left and right sequence context, which are forward and backward layer. We take the word representation $X = \{x_1, x_2, ..., x_n\}$ as input, at the time step t the LSTM units could be demonstrated:

$$i_t = \sigma\left(W_i w_t + U_i h_{t-1} + b_i\right) \tag{4}$$

$$f_t = \sigma\left(W_f w_t + U_f h_{t-1} + b_f\right) \tag{5}$$

$$o_t = \sigma\left(W_o w_t + U_o h_{t-1} + b_o\right) \tag{6}$$

$$\widetilde{c_t} = \tan h\left(W_c w_t + U_c h_{t-1} + b_c\right) \tag{7}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \widetilde{c_t} \tag{8}$$

$$h_t = o_t \odot \tan h\left(c_t\right) \tag{9}$$

where i_t, f_t, and o_t are input gate, forget gate and output gate, respectively. The parameters W_i, U_i represent the weight matrix of the input gate i_t. The parameters W_f, U_f represent the weight matrix of the forget gate f_t. The parameters W_o, U_o represent the weight matrix of the output gate o_t. The parameters b_i, b_f, and b_o are bias vectors of input gate, forget gate, and output gate, respectively. The parameters W_c and U_c are the weight matrix of new memory content $\widetilde{c_t}$. The b_c is the bias vector of the new memory content $\widetilde{c_t}$. The h_t is an LSTM hidden state, c_t is the current cell state, $\odot$ denotes element-wise multiplication and tanh is a hyperbolic tangent function.

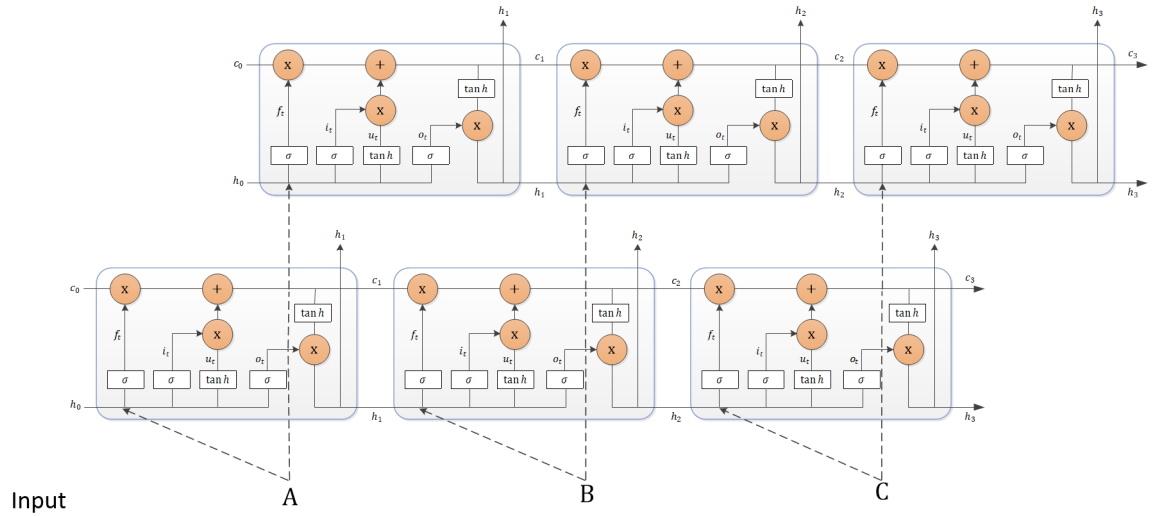

Figure 4. The whole architecture of our proposed MALNet model.

At the time step t, the output of the i$^t h$ word is shown in the following equation:

$$h_t = \left[\overrightarrow{h_t} \oplus \overleftarrow{h_t}\right] \tag{10}$$

where $\overrightarrow{h_t}$, $\overleftarrow{h_t} \in \mathbb{R}^{d^h}$ are the hidden states of the forward and backward LSTM at time step t. $h_t \in \mathbb{R}^{2d^h}$ represents the connection of the hidden states at time step t. d^h is the hidden size of BLSTM. The $\oplus$ is the connection operator. Then we separate two entity vectors h_t^{e1} and h_t^{e2} from the BLSTM output vector, which can represent the tagged entity context information at the time step t.

Due to the uncertainty of sentence length and the distance between entities [33,34], although BLSTM can capture context information, it performs poorly in long texts. To solve this problem, we add an attention layer after the BLSTM layer, which can capture longer texts effectively. In attention part, we construct Q is the vector h_t^{e1} and h_t^{e2}, K,V is the output of the BLSTM. To maintain dimensional consistency, extend h_t^{e1} and h_t^{e2} to the same dimension as BLSTM. The scaled dot-product attention is calculated by the following equation:

$$\text{Entity} - \text{Att}(\text{Q}, \text{K}, \text{V}) = \text{softmax}\left(\frac{\text{QK}^{\text{T}}}{\sqrt{\text{d}_l}}\right)\text{V} \tag{11}$$

$$\text{Entity} - \text{head}_{\text{i}} = \text{Entity} - \text{Att}\left(W_i^Q\text{Q}, W_i^K\text{K}, W_i^V\text{V}\right) \tag{12}$$

$$\text{H}_\text{i} = W^C\left[(\text{Entity} - \text{head}_1) \oplus \ldots \oplus (\text{Entity} - \text{head}_\text{r})\right] \tag{13}$$

where $W_i^Q, W_i^K, W_i^V \in \mathbb{R}^{d_l * d_l}$ are the projections matrices, $W^C \in \mathbb{R}^{d_l * r}$ is a mapping parameter from input space to representation space, r is the number of the attention heads, $d_l = 2 * d^h$ represent the dimension of the BLSTM output, H_i represents the entity i with BLSTM output attention results. By the attention mechanism, we can get the output H_1 and H_2.

(3) Sentence Level Attention Filter Layer

In recent years, attention mechanism used to learn text classification [35], question answering [36], and named entity recognition [37]. The words in the sentence contain different levels of importance [38]. To effectively distinguish the valid features and the invalid, we do some fine-tuning of attention mechanisms to build an attention filter layer. Firstly, we concatenate the two channel's output in the attention-based BLSTM layer. The formula is as follows:

$$\mathrm{T} = [\mathrm{H}_1 \oplus \mathrm{Z} \oplus \mathrm{H}_2] \tag{14}$$

The $\oplus$ represents the connection operator. Then we introduced the latent entity types [29]. It may be more than one relationship between a particular entity and other entities. We use LET (Latent Entity Types) to extract the latent types to improve ability to extract relationships. The mathematical formulation is the follows:

$$Y_i = \mathrm{softmax}(\mathrm{W} * \mathrm{H})\mathrm{H} \tag{15}$$

where $\mathbf{W} \in \mathbb{R}^{(2d^h)*h}$ represent the LET weight matrix, h is the hyper parameter latent type. H is the vector h_t^{e1} and h_t^{e2}. Traditional models take all position embeddings as a part of the word presentation, which may cause noise amplification and influence the original word representation [19,28]. In this paper, we introduce it into the filter layer. The position embeddings of the ith word is encoded as p_i^{e1}, $p_i^{e2} \in \mathbb{R}^{dp}$, where d^p is the dimension of the position embeddings. For the sentence, the position embeddings can be represented as $p_1, p_2 \in \mathbb{R}^{dp*n}$, n is the number of the words.

All high-level features about this sentence may affect the classification. Based on this idea, we take these features into consideration as shown in Figure 5: (1) the latent entity type vector Y_1 and Y_2; (2) the position embeddings of the sentence which reflect each word relative to entities p_1 ,p_2; (3) the concatenation sum in the attention-based BLSTM layer. We construct a high-level feature selection matrix C, which consists of all features. $\mathrm{C} = [\mathrm{T} \oplus \mathrm{Y}_1 \oplus \mathrm{Y}_2 \oplus p_1 \oplus p_2]$, $\mathrm{C} \in \mathbb{R}^{\left(d^w+2d^h+2d^p\right)}$. The $\oplus$ represents the connection operator. Finally, in order to better solve sentences of different length, we introduce advanced features in self-attention, but they only participate in the selection of factors that determine the relationship between sentences, not be part of the output. The representation R of the sentence can be calculated by the following equation:

$$\mathrm{M} = \tanh\left(\mathrm{W_w C} + \mathrm{b_w}\right) \tag{16}$$

$$\alpha = \mathrm{softmax}\left(\mathrm{W_h M}\right) \tag{17}$$

$$\mathrm{R} = \alpha \odot \mathrm{T} \tag{18}$$

where $\mathrm{W}_w \in \mathbb{R}^{\left(d^w+2d^h+2d^p\right)*e}$, e represents the attention size. $\mathrm{R} \in \mathbb{R}^{\left(d^w+2d^h+2d^p\right)}$, $\odot$ denotes element-wise multiplication. $\mathrm{W_h} \in$ e represents a transpose vector. α represents the weight of the filter layer. By this layer, we get the high-level features.

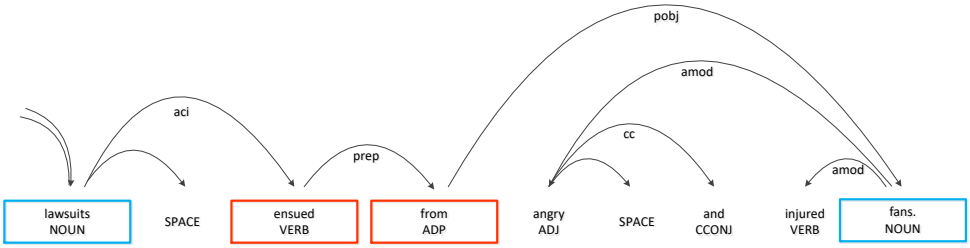

Figure 5. The whole architecture of the sentence level attention filter layer.

(4) Key Word Layer

Shortest dependency path (SDP) to mark important words in long sentences and give key words with highlight weights. However, we find that the key word assigned by SDP cannot really express the core meaning of a sentence and not all sentences can extract key words through SDP.

As shown in Figure 6, the blue box represents the target word, and the red box represents the word on SDP between two words. It is obvious that the word "injured" is the predicate which connect "lawsuits" and "fans". However, SDP think "ensued" and "from" is the key word and assign them a high weight value, but in this sentence, "injured" is also a key word, which ignored by SDP.

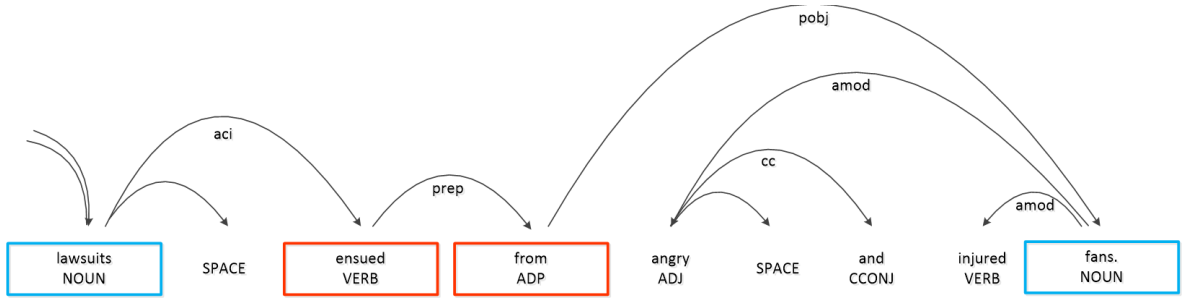

Figure 6. An example of shortest dependency path (SDP) extracting key word.

Besides, not all sentences can obtain key words by SDP. We tested 8000 sentences in the training set of SemEval-2010 task 8 dataset and find that there are 16 sentences without SDP between the two target words. It means that the operation of giving high weight fails in some sentences, and all words have the same effect in classification, which cannot highlight the importance of key words.

In order to better extract the key words, we make a semantic rule according to the task of relationship classification. Because the words between the two target words basically cover all the possible information, we take out all the words. According to the task of relation classification, we have designed a rule: (1) remove all parallel words such as "and", "or" and " but"; (2) remove all adverbial words based on part of speech tagging, keep all actions between two words, and provide effective information for classification; (3) according to the above rule, the core words of sentences we have processed are as shown in the Figure 7.

Figure 7. An example of our rule extracting key words.

We extract key words from each sentence by the rule, and then splice their word vectors together to reduce the distance between them. Because the length of the spliced sentence is smaller than that of the original sentence, we use a convolution neural network (CNN) directly.

The advantage of a convolution neural network is that it can extract local features more effectively all of the key words of the sentence S are then represented as a list of vectors $(X_0, X_1, \cdots, X_i)$, where xi corresponds to the key word. As shown in Figure 8, with a slide window of size k, the CNN can extract local features between key words.

$$R^* = \text{relu}\ (W_{CNN}S + b_{CNN}) \tag{19}$$

where $S \in \mathbb{R}^{d^w * i}$,$W_{CNN} \in \mathbb{R}^{d^w * k}$ is a weight matrix with a channel size K. Through the CNN operation of core vocabularies, we can extract features to provide assistance for our classification.

(5) Classification Layer

The features extracted by Key Word Layer will help the classification process, so we add the output of the key word layer and the Sentence Level Attention Filter Layer. α represents weight

coefficient. We obtain a high-level sentence representation for the relationship, which we can directly use to predict the label $\hat{y}$. This classification process consists of a softmax classifier and the probability $\hat{p}(y \mid S, \theta)$ is:

$$\hat{p}(y \mid S, \theta) = \text{softmax}\left(\left(W^{s}R + b^{s}\right) + \alpha\left(W^{*}R^{*} + b^{*}\right)\right) \tag{20}$$

where y is a target relation class and S represent the sentence. R represents the output of the Attention Filter layer. The θ parameter represents the whole trainable parameters in the whole network. The relation label with highest probability value is identified as ultimate result:

$$\hat{y} = \arg\max_{y} \hat{p}(y \mid S, \theta) \tag{21}$$

For the propose of making a clear distinction, we made some adjustments based on the ranking loss function [18]. The formula is as follows:

$$L = \log\left(1 + \exp\left(\gamma\left(m^{+} - s_{\theta}(x)_{y^{+}}\right)\right)\right) + \log\left(1 + \exp\left(\gamma\left(m^{-} + s_{\theta}(x)_{c^{-}}\right)\right)\right) + \lambda\|\theta\|_{2}^{2} \tag{22}$$

where y^{+} represents the correct label and c^{-} represent the highest probability sample among all incorrect relation types. $s_{\theta}(x)_{y^{+}}$ and $s_{\theta}(x)_{c^{-}}$ represent the softmax score of the correct relation label and the negative category chose with the highest probability among all incorrect relation types. The λ is a L2 regularization hyper parameter. In our experiment, we set γ to 2.0, m^{+} to 1.0, and m^{-} to 0.0. When the loss function decreases, the first term in the right side $s_{\theta}(x)_{y^{+}}$ increases and the second term in the right side $s_{\theta}(x)_{c^{-}}$ decreases. Compared with the cross-entropy loss function, this loss function can better distinguish positive labels from negative labels and make the boundary larger. We introduced the L2 regularization to avoid overfitting and improve generalization ability. In the training phrase, we optimize the loss function by using Adadelta algorithm.

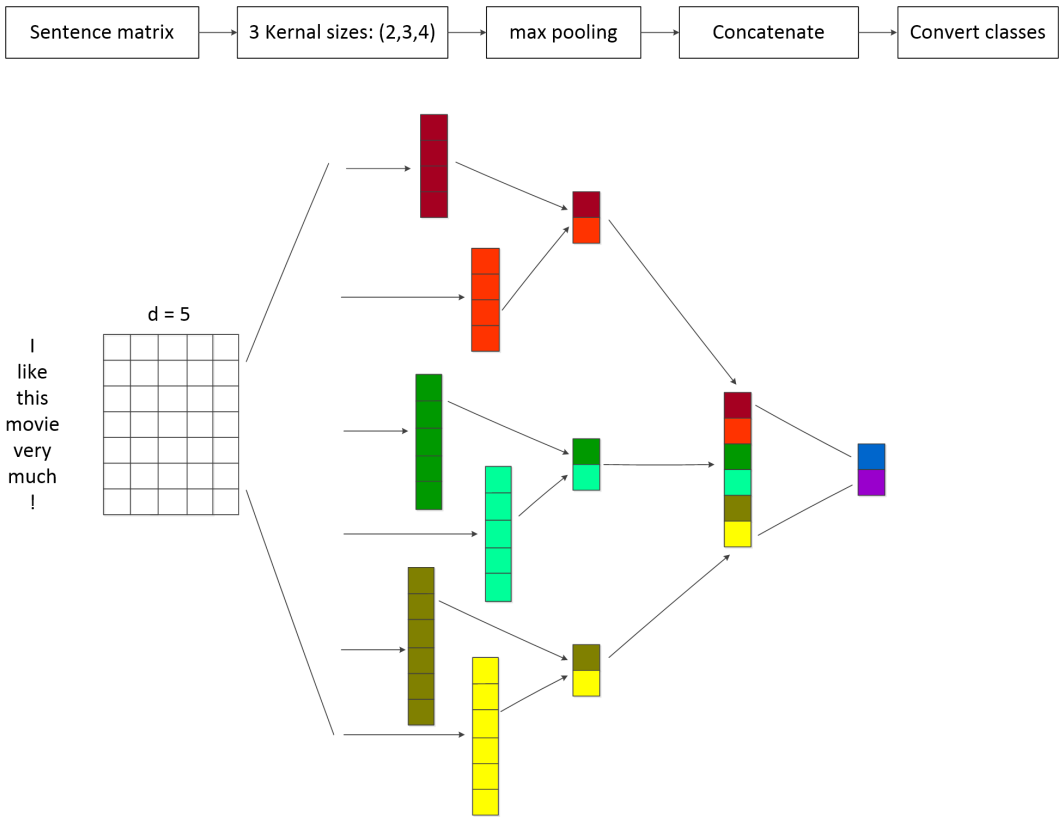

Figure 8. The Key Word Layer network structure.

4. Experiment

To evaluate the effectiveness of our MALNet model, we conducted an experiment on the SemEval-2010 Task 8 dataset and KBP-37 dataset. Compared with the other methods, our model performs better than them.

4.1. Datasets

We evaluated our model on two datasets, including SemEval-2010 Task8 dataset and KBP-37 dataset. In order to analyze the difference between the datasets, we counted the lengths of sentences and the distance between two tagged entities. The statistical features are shown in Figures 9 and 10. We can see that in SemEval-2010 Task8 dataset, around 98% of entities' distance is less than 15 and around 85% sentences length are less than 30, which means that most of the datasets are short sentences. On the contrary, KBP-37 dataset contains a large number of long sentences and the entities distance greater than 15 reached a quarter. In the Figure 10, we can see that SE focuses on short sentences and KBP on long sentences. Figure 11 shows the statistical properties of these two datasets.

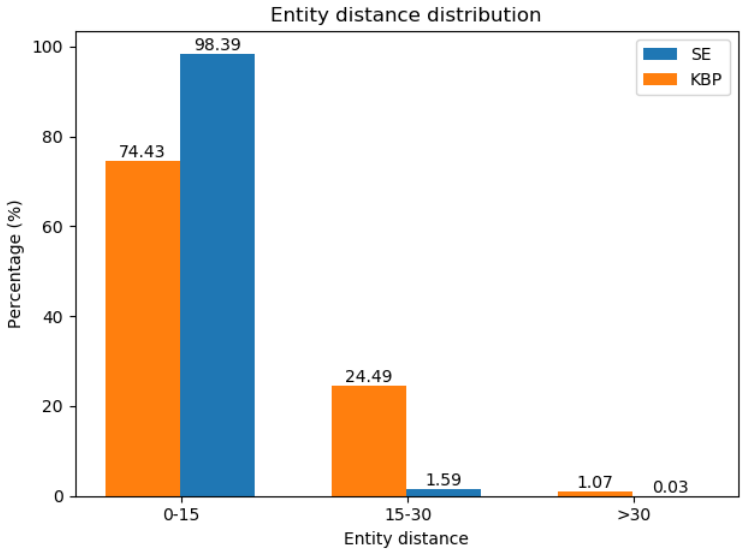

Figure 9. The entity distance distribution.

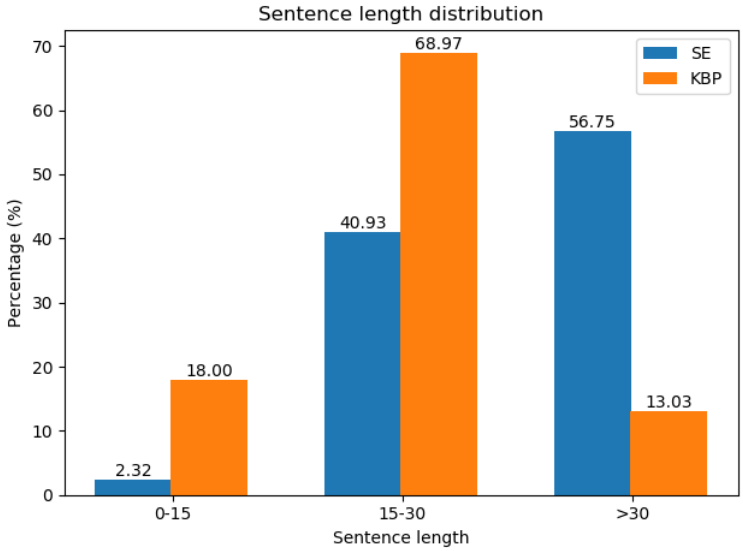

Figure 10. The sentence length distribution.

SemEval-2010 Task8 dataset is a commonly used benchmark for relation classification [14]. The dataset total number of relations is 19. There are 10,717 annotated sentences that consist of 8000 samples for training and 2717 samples for testing. We used the Macro-F1 score (excluding "other").

KBP-37 is a dataset that contains more specific entities and relations [17]. In this dataset, there are more entities are names of persons, organizations or cities, which means more noise in the sentences. The total dataset number of relations is 37. In addition, there are more long sentences and more entities are name organizations and place names in the KBP-37 dataset. It means that the unseen words will affect classification accuracy.

Dataset	SemEval-2010 Task8	KBP-37
Number of training data	8000	17,641
Number of test data	2717	3405
Number of relation types	19	37

Figure 11. The statistical properties of these two datasets.

4.2. Experiment Settings

In our experiment, we set the word embeddings size d^w to 1024. The dimension of the position embeddings d^p were set to 50. The batch size were set to 20. To avoid over-fitting, we applied a dropout on the word embeddings layer, the output of the BLSTM, and self-attention filter layer. The dropout rate was 0.3, 0.3, and 0.5, respectively. In the BLSTM layer, we set the LSTM hidden size to 300. We set the learning rate to 0.1 and decay rate to 0.9. The regularization parameter λ was set to 0.001. The attention heads were set to 4. In the key word layer, we set the window size k to 2, 3, 4, and the number of convolution kernels to 50. In the SemEval-2010 dataset, we set the weight coefficient α to 1 and the KBP dataset to 0.2.All the hyper parameter are shown in Table 1.

Table 1. The hyper parameter of the dataset.

Parameters	Values
word emb dim	1024
word dropout	0.3
pos emb dim	50
LSTM dropout	0.3
LSTM hidden size	300
batch size	20
multi-heads	4
dropout	0.5
Regularization	0.001
learning rate	0.1
latent_type	3
decay rate	0.9
att_size	50

4.3. Experiment Results

We compared our model to the previous model on both dataset SemEval-2010 Task 8 and KBP-37. Tables 2 and 3 describe the performance of our model and other methods in the datasets.

In the SemEval-2010 Task 8 dataset, we compared our model with the baseline methods CNN and RNN. Nguyen et al. [20] proposed the perspective-CNN network applied to the relation classification task. They tested it on multiple types of convolution kernels and finally achieved an F1-score of 82.8%. Due to the limitation of convolution kernel size, Zhou et al. [15] introduced attention mechanism into a BLSTM to extract sentence features more effectively. They achieved 84%, which is the state-of-the-art result. Dos Santos et al. [18] constructed a classification by ranking CNN (CR-CNN) to tackle the relation classification task. They improved the loss function on the basis of CNN. Adilova et al. [34] proposed a supervised ranking CNN model, which is an improvement on CRCNN. They tested their

model on the dataset and achieved 84.39%. Zhang et al. [27] proposed a RCNN model that combines RNN and CNN in the network structure. This model got a F1-score 83.7%. Cao et al. [28] applied adversarial training into the BLSTM model and achieved 83.6%. Zhang et al. [39] proposed a model named BiLSTM-CNN (Bi-directional Long Short-Term Memory- Convolutional Neural Networks) and added the position embeddings into the input. J Lee et al. [29] applied latent-attention into BLSTM and achieved 85.2%. G. Tao et al. [40] proposed subsequence-level entity attention LSTM and achieved 84.7%. Our MALnet outperforms the previous methods and achieved 86.4% in this dataset. In order to reflect the effective filtering effect of the filter layer, we designed a series of comparative experiment. We removed the Attention Filter Layer and key word layer from origin MALnet and the FI-score is decreased to 84.3%. When we just remove the filter layer our network achieved 85.6%. When we just remove the key word layer our network achieved 85.1%.

Table 2. Experiment results on SemEval-2010 Task 8 dataset.

Model	F1
Perspective -CNN (Nguyen et al. 2015)	82.8
ATT-RCNN (Zhang et al. 2018)	83.7
ATT-BLSTM (Zhou et al. 2016)	84
CRCNN (dos Santos et al. 2015)	84.1
BLSTM+ATT+AT+GATE (Cao et al. 2018)	84.3
Supervised Ranking CNN (Adilova et al. 2018)	84.39
BiLSTM-CNN+PI (Zhang et al. 2018)	82.1
BiLSTM-CNN+PF+PI (Zhang et al. 2018)	83.2
Subsequence-Level Entity Attention LSTM (G.Tao et al. 2019)	84.7
Entity-ATT-LSTM (J Lee et al. 2019)	85.2
MALNet (without filter layer and key word layer)	84.3
MALNet (without filter layer)	85.6
MALNet (without key word layer)	85.1
MALNet	86.3

In order to reflect the generalization ability of our model in different datasets, we also used the same methods to compare with our model in KBP-37 dataset. The Perspective-CNN does not perform very well in a KBP-37 dataset for its limitation of the convolution kernel in long sentences. Models using an RNN performed better than CNN in long sentences. The Supervised Ranking CNN [34] achieved 61.26% in this dataset. The proposed method Entity-ATT-LSTM achieves 58.1% in this dataset. In the experiment for the KBP dataset, we obtained an F1-score of 61.4%. We also removed the Attention Filter Layer from the original MALnet and the FI-score then decreased to 58.3%, which showed the effectiveness of the filter layer. When we just removed the filter layer our network achieved 59.3%. When we just removed the key word layer, our network achieved 60.1%.

Compared with the reference model, we can conclude that our model is more robust and has good adaptability in short and long sentences. It can also be seen from the comparative experiment that the module we designed is also effective in relation classification.

Table 3. Experiment results on KBP-37 dataset.

Model	F1
Perspective -CNN (Nguyen et al. 2015)	57.1
ATT-RCNN (Zhang et al. 2018)	59.4
ATT-BLSTM (Zhou et al. 2016)	58.8
CRCNN (dos Santos et al. 2015)	58.5
Supervised Ranking CNN (Adilova et al. 2018)	61.26
BiLSTM-CNN+PI (Zhang et al. 2018)	59.1
BiLSTM-CNN+PF+PI (Zhang et al. 2018)	60.1
Entity-ATT-LSTM (J Lee et al. 2019)	58.1
MALNet (without filter layer and key word layer)	58.3
MALNet (without filter layer)	59.3
MALNet (without key word layer)	60.1
MALNet	61.4

5. Conclusions

In this paper, we propose a novel neural network model MALNet for relation classification. Our model uses raw text with word embeddings and position embeddings as input. To extract primary features, we designed an attention-based BLSTM layer. By this layer, the semantic information is transformed into advanced features. Then, we construct a Sentence Level Filter Layer to preserve features that facilitate classification effectively. In the current research, we found that there are some noises when using external tools for syntax analysis, so we construct a semantic rule to solve this problem, and extract key words to construct a key word layer to help the relationship classification task. Our experiment was carried out on two datasets: one has mostly short sentences, the other has mostly long sentences. In order to highlight the importance of the filter module and key word layer, we also made some comparative experiments. The experiment results show that our MALNet model works better than previous state-of-the-art methods on both the SemEval-2010 Task 8 dataset and KBP-37 dataset and the module we designed proved to be effective in relation classification.

In the future, we will consider introducing the Named Entity Recognition (NER) into relation classification. The entity recognition can mark out the nouns in a sentence, which can cooperate with the classification network to refine the sentence structure.

Author Contributions: D.W. and Y.J. conceived and designed the experiments; D.W. wrote the first draft; Y.J. and W.G. revised the article; D.W. and Y.J. analyzed the data. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the NSFC, China, under Grant 61771299, in part by the National Key Research and Development Program of China under Grant 2018YFB2101303, in part by the Key Laboratory of Specialty Fiber Optics and Optical Access Networks, Shanghai University, under Grant SKLSFO2012-14, in part by the Key Laboratory of Wireless Sensor Network and Communication, Shanghai Institute of Microsystem and Information Technology, in part by the Shanghai Education Committee, Chinese Academy of Sciences, in part by the Shanghai Science Committee under Grant 19511102803, in part by the H2020 under Grant 778305, and in part by the Innovate U.K. under Grant 10734.

Acknowledgments: The authors are very grateful to three anonymous referees for their valuable comments on the paper, which have considerably improved the presentation of this paper.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Fader, A.; Soderland, S.; Etzioni, O. Identifying relations for open information extraction. In Proceedings of the Empirical Methods Natural Language Processing (EMNLP), Stroudsburg, PA, USA, 27–31 July 2011; pp. 1535–1545.
2. Banko, M.; Cafarella, M.J.; Soderland, S.; Broadhead, M.; Etzioni, O. Open information extraction from the web. In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), Hyderabad, India, 6–12 January 2007; pp. 2670–2676.
3. Hao, Y.; Zhang, Y.; Liu, K.; He, S.; Liu, Z.; Wu, H.; Zhao, J. An end-to-end model for question answering over knowledge base with cross-attention combining global knowledge. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL), Vancouver, BC, Canada, 30 July–4 August 2017; pp. 221–231.
4. Sorokin, D.; Gurevych, I. Context-aware representations for knowledge base relation extraction. In Proceedings of the Empirical Methods Natural Language Processing (EMNLP), Copenhagen, Denmark, 7–11 September 2017; pp. 1784–1789.
5. Sun, R.; Jiang, J.; Fan, Y. Using syntactic and semantic relation analysis in question answering. In Proceedings of the 14th Text REtrieval Conference (TREC), Gaithersburg, MD, USA, 15–18 November 2005; p. 243.
6. Yih, W.-T.; He, X.; Meek, C. Semantic parsing for single-relation question answering. In Proceedings of the Association for Computational Linguistics (ACL), Baltimore, MD, USA, 22–27 June 2014; pp. 643–648.
7. Verga, P.; Belanger, D.; Strubell, E.; Roth, B.; McCallum, A. Multilingual Relation Extraction Using Compositional Universal Schema. Available online: https://arxiv.gg363.site/abs/1511.06396 (accessed on 18 July 2020).

8. Minard, A.-L.; Ligozat, A.L.; Grau, B. Multi-class SVM for Relation Extraction from Clinical Reports. In Proceedings of the Recent Advances in Natural Language Processing, RANLP 2011, Hissar, Bulgaria, 12–14 September 2011; pp. 604–609.
9. Jiang, J.; Zhai, C. A systematic exploration of the feature space for relation extraction. In Proceedings of the Human Language Technologies: The Conference of the North American Chapter of the Association for Computational Linguistics, Rochester, NY, USA, 22–27 April 2007; pp. 113–120.
10. Yang, Y.; Tong, Y.; Ma, S.; Deng, Z.-H. A position encoding convolutional neural network based on dependency tree for relation classification. In Proceedings of the Empirical Methods Natural Language Processing (EMNLP), Austin, TX, USA, 1–5 November 2016; pp. 65–74.
11. Dai, Y.; Guo, W.; Chen, X.; Zhang, Z. Relation classification via LSTMs based on sequence and tree structure. *IEEE Access* **2018**, *6*, 64927–64937. [CrossRef]
12. Liu, Y.; Wei, F.; Li, S.; Ji, H.; Zhou, M.; Wang, H. A Dependency-Based Neural Network for Relation Classification. July 2015. Available online: https://arxiv.gg363.site/abs/1507.04646 (accessed on 10 July 2020).
13. Xu, J.; Wen, J.; Sun, X.; Su, Q. A Discourse-Level Named Entity Recognition and Relation Extraction Dataset for Chinese Literature Text. June 2019. Available online: https://arxiv.org/abs/1711.07010 (accessed on 13 July 2020).
14. Rink, B.; Harabagiu, S. UTD: Classifying semanticrelations bycombining lexical and semantic resources. In Proceedings of the 5th International Workshop on Semantic Evaluation, Los Angeles, CA, USA, 15–16 July 2010; pp. 256–259.
15. Zhou, P.; Shi, W.; Tian, J.; Qi, Z.; Li, B.; Hao, H.; Xu, B. Attention-based bidirectional long short-term memory networks for relation classification. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL), Berlin, Germany, 7–12 August 2016; pp. 207–212.
16. Zhang, S.; Zheng, D.; Hu, X.; Yang, M. Bidirectional long short-term memory networks for relation classification. In Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation (PACLIC), Shanghai, China, 30 October–1 November 2015; pp. 73–78.
17. Zhang, D.; Wang, D. Relation Classification via Recurrent Neural Network. December 2015. Available online: https://arxiv.org/pdf/1508.01006.pdf (accessed on 18 June 2020).
18. Dos Santos, C.N.; Xiang, B.; Zhou, B. Classifying relations by ranking with convolutional neural networks. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL), Beijing, China, 24 May 2015; Volume 1, pp. 626–634.
19. Xiao, M.; Liu, C. Semantic relation classification via hierarchical recurrent neural network with attention. In Proceedings of the COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, Osaka, Japan, 11–16 December 2016; pp. 1254–1263.
20. Nguyen, T.H.; Grishman, R. Relation extraction: Perspective from convolutional neural networks. In Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing, Denver, CO, USA, 5 June 2015; pp. 39–48.
21. Zeng, D.; Liu, K.; Lai, S.; Zhou, G.; Zhao, J. Relation classification via convolutional deep neural network. In Proceedings of the COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, Dublin, Ireland, 23–29 August 2014; pp. 2335–2344.
22. Ma, D.; Li, S.; Zhang, X.; Wang, H. Interactive attention networks for aspect-level sentiment classification. In Proceedings of the International Joint Conferences on Artificial Intelligence (IJCAI), Melbourne, Australia, 19–25 August 2017; pp. 4068–4074.
23. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems (NIPS), Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
24. Hochreiter, S.; Schmidhuber, J. LSTM can solve hard long time lag problems. In Proceedings of the Advances in Neural Information Processing Systems, Cambridge, MA, USA, December 1996; p. 473.
25. Yan, X.; Mou, L.; Li, G.; Chen, Y.; Peng, H.; Jin, Z. Classifying Relations via Long Short Term Memory Networks along Shortest Dependency Path. *arXiv* **2015**, arXiv:1508.03720.
26. Shen, Y.; Huang, A. Attention-based convolutional neural network for semantic relation extraction. In Proceedings of the 26th International Conference on Computational Linguistics: Technical Papers (COLING 2016), Osaka, Japan, 11–16 December 2016; pp. 2526–2536.

27. Zhang, X.; Chen, F.; Huang, R. A combination of RNN and CNN for attention-based relation classification. *Procedia Comput. Sci.* **2018**, *131*, 911–917. [CrossRef]
28. Cao, P.; Chen, Y.; Liu, K.; Zhao, J. Adversarial Training for Relation Classification with Attention Based Gate Mechanism. In Proceedings of the China Conference on Knowledge Graph and Semantic Computing (CCKS), Tianjin, China, 14–18 August 2018; pp. 91–102.
29. Lee, J.; Seo, S.; Choi, Y.S. Semantic Relation Classification via Bidirectional Lstm Networks with Entity-Aware Attention Using Latent Entity typing June 2019. Available online: https://www.mdpi.com/2073-8994/11/6/785 (accessed on 27 July 2020).
30. Bengio, Y.; Ducharme, R.; Vincent, P.; Jauvin, C. A Neural Probabilistic Language Model. *J. Mach. Learn. Res.* **2003**, *3*, 137–1155.
31. Peters, M.E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; Zettlemoyer, L. Deep contextualized word representations. *arXiv* **2018**, arXiv:1802.05365.
32. Zhu, Z.; Su, J.; Zhou, Y. Improving Distantly Supervised Relation Classification with Attention and Semantic Weight. *IEEE Access* **2019**, *7*, 91160–91168. [CrossRef]
33. Roze, C.; Braud, C.; Muller, P. Which Aspects of Discourse Relations Are Hard to Learn? September 2019. Available online: https://www.aclweb.org/anthology/W19-5950/ (accessed on 7 June 2020).
34. Adilova, L.; Giesselbach, S. Making efficient use of a domain expert's time in relation extraction. *arXiv* **2018**, arXiv:1807.04687.
35. Du, J.; Gui, L.; He, Y.; Xu, R.; Wang, X. Convolution-based neural attention with applications to sentiment classification. *IEEE Access* **2019**, *7*, 27983–27992. [CrossRef]
36. Wang, P.; Wu, Q.; Shen, C.; Dick, A.; van den Hengel, A. Fvqa: Fact-based visual question answering. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 2413–2427. [CrossRef] [PubMed]
37. Jin, Y.; Xie, J.; Guo, W.; Luo, C.; Wu, D.; Wang, R. LSTM-CRF Neural Network with Gated Self Attention for Chinese NER. *IEEE Access* **2019**, *7*, 136694–136709. [CrossRef]
38. Sun, J.; Li, Y.; Shen, Y.; Ding, W.; Shi, X.; Zhang, L.; Shen, X.; He, J. Joint Self-Attention Based Neural Networks for Semantic Relation Extraction. *J. Inf. Hiding Priv. Prot.* **2019**, *1*, 69. [CrossRef]
39. Zhang, L.; Xiang, F.; Tan, Y.; Shi, Y.; Tang, Q. Relation classification via BiLSTM-CNN. In Proceedings of the International Conference on Data Mining and Big Data, Shanghai, China, 17–22 June 2018; pp. 373–382.
40. Tao, G.; Gan, Y.; He, Y. Subsequence-Level Entity Attention LSTM for Relation Extraction. In Proceedings of the 2019 16th International Computer Conference on Wavelet Active Media Technology and Information Processing, Chengdu, China, 14–15 December 2020.