RESEARCH PAPER

Relation Extraction Based on Prompt Information and Feature Reuse

Ping Feng[1,2,3†], Xin Zhang[2], Jian Zhao[2,3], Yingying Wang[2], Biao Huang[2]

[1]Jilin University, Changchun Jilin 130012, China

[2]Changchun University, Changchun Jilin 130022, China

[3]Jilin Provincial Key Laboratory of Human Health State Identification and Function Enhancement, Changchun Jilin 130022, China

Keywords: relation extraction; language model; prompt information; feature reuse; loss function

Citation: Feng, P., Zhang, X., Zhao, J. et al.: Relation extraction based on prompt information and feature reuse. Data Intelligence 5(3), 817-833 (2023). doi: 10.1162/dint_a_00192

Received: Feburary 20, 2023; Revised: March 14, 2023; Accepted: April 18, 2023

ABSTRACT

To alleviate the problem of under-utilization features of sentence-level relation extraction, which leads to insufficient performance of the pre-trained language model and underutilization of the feature vector, a sentence-level relation extraction method based on adding prompt information and feature reuse is proposed. At first, in addition to the pair of nominals and sentence information, a piece of prompt information is added, and the overall feature information consists of sentence information, entity pair information, and prompt information, and then the features are encoded by the pre-trained language model ROBERTA. Moreover, in the pre-trained language model, BIGRU is also introduced in the composition of the neural network to extract information, and the feature information is passed through the neural network to form several sets of feature vectors. After that, these feature vectors are reused in different combinations to form multiple outputs, and the outputs are aggregated using ensemble-learning soft voting to perform relation classification. In addition to this, the sum of cross-entropy, KL divergence, and negative log-likelihood loss is used as the final loss function in this paper. In the comparison experiments, the model based on adding prompt information and feature reuse achieved higher results of the SemEval-2010 task 8 relational dataset.

† Corresponding author: Ping Feng (E-mail: fengping@ccu.edu.cn; ORCID: 0000-0003-4865-1454).

1. INTRODUCTION

Relation extraction, as a basic information extraction task, aims to identify the relationship between pairs of nominals in a given sentence from a set of predefined relationships of interest. The work process can be briefly summarized as follows: the triple r(e1, e2) is extracted from the unstructured text. Where e1 and e2 are entities in the utterance, generally nouns or phrases formed by nouns, and r denotes the relationship between entities e1 and e2.

Relation extraction plays a crucial role in natural language processing applications that require a relational understanding of the unstructured text, such as question answering the application, recommendation algorithm, semantic search, knowledge base filling, and knowledge graph construction. Many tasks of natural language processing can benefit from accurate relation classification. Therefore, relation extraction has attracted a lot of attention. The common approach nowadays is to fine-tune pre-trained language models such as BERT [1], ROBERTA [2], and GPT [3], etc. to achieve relation classification. The existing sentence-level relation extraction is also mainly based on the language model with various innovations. However, in the process of fine-tuning the language model to the relation extraction task, the insufficient feature selection makes the language model too fine-tuned to the downstream task, thus not giving full play to the performance of the language model; at the same time, the model does not make sufficient use of the feature vector.

To this end, this paper proposes a sentence-level relation extraction method based on adding prompt information and feature reuse. The modification of the sentence is shown in Table 1. This method first adds a prompt message in addition to the original sentence-level features and entity-pair features: "What is the relationship between entity one (e1) and entity two (e2) in the above sentence?". Then the sentence features, entity pair features, and prompt features are all encoded by ROBERTA [2]. The encoded data is then fed into the model, and in the model composition this paper chooses the ROBERTA [2] language model as a basis for the overall model, and BIGRU is introduced in the process of model fine-tuning, from which another feature is constructed. In the hidden layer of the model, five features are proposed in this paper noted as $Feature_{cls}$, $Feature_{bigru}$, $Feature_{entity1}$, $Feature_{entity2}$, $Feature_{prompt}$. Finally, feature reuse is performed to form four different outputs, and ensemble-learning soft voting is used for the output Voting is performed and the voted results are used for predictive relation classification. The main contribution of this method is to add prompt information to the relation extraction task, which not only solves the problem of insufficient feature information but also allows the model to give full play to its performance; secondly, feature reuse and ensemble-learning are used to solve the problem of insufficient utilization of feature vectors and further improve the robustness of sentence-level relation extraction results; finally, in this paper, the same batch of data is fed into the model twice before and after finally, the same batch of data is fed into the model twice to obtain two different distributions, and a new loss function consisting of the cross-entropy, KL divergence, and negative log-likelihood loss of these two distributions is used to optimize the model.

Table 1. Processing of statements in a dataset.

Sentence (1)	The <e1> legend </e1> was derived from a much older <e2> publication </e2>.
Relationship	Entity-Origin (e1, e2)
Modification	The $ legend $ was derived from a much older # publication #. @ what is the relationship between legend and publication in the above sentence? @
Sentence (2)	Most <e1> deaths </e1> from the accident were caused by radiation <e2> poisoning </e2>.
Relationship	Cause-Effect (e2, e1)
Modification	Most $ deaths $ from the accident were caused by radiation # poisoning #. @ what is the relationship between deaths and poisoning in the abovesentence? @
Sentence (3)	The <e1> leftovers </e1> are pushed into the <e2> colon </e2>.
Relationship	Entity-Destination (e1, e2)
Modification	The $ leftovers $ are pushed into the # colon #. @ what is the relation-ship between leftovers and colon in the above sentence? @

2. RELATED WORK

The task of relation classification is a very important part of the knowledge graph construction process. The methods of relation extraction, in general, include unsupervised relation classification and supervised relation classification, with supervised relation classification, which is usually considered a multiclassification problem. The performance of traditional relation classification depends mainly on the quality of features, but errors often occur during feature extraction with NLP tools, reducing the overall performance of the model. To solve this problem of feature extraction errors, Zeng et al. [4], Zheng et al. [5], Zheng et al. [6] successively proposed the use of convolutional neural networks, recurrent neural networks, and graph neural networks for relationship extraction. Although these neural networks can encode and convert entity pairs and sentence information into feature vectors, which provides some improvement in model performance, these approaches do not take into account which information in the sentence is more important. For this reason, Shen et al. [7], Zhou et al. [8], Guo et al. [9] proposed models for neural networks with attention mechanisms, which were added to convolutional neural networks, recurrent neural networks, and graph neural networks, respectively, to further improve the performance of relational extraction models. On top of this, Lee et al. [10] added the perception of entities to enhance the robustness of the attention mechanism.

After the emergence of language models EMLO [11], BERT [1], GPT [3], etc., language models were widely used for relation extraction tasks. Alt et al. [12] proposed a new approach based on Transform [13] architecture for relation extraction, using pre-learned implicit language features combined with Transform. Wang et al. [14] applied BERT applied to relation extraction and used an entity-aware self-attention mechanism to inject relation information related to multiple entities in each layer of the hidden state to achieve the prediction of multiple relations by encoding them once. Wu et al. [15] similarly proposed a relationship extraction model based on BERT, while encoding the information of entity pairs into the feature vector as well, thus effectively improving the performance of the model. Tian et al. [16] employ a graph convolutional neural network based on the attention mechanism on top of the BERT encoding, which can

better parse the information in the dependency tree. Tao et al. [17] extract syntactic indicators guided by syntactic knowledge and then encode them using language models, which mitigates the noise of the data. Han et al. [18] instead propose prompt-based learning that converts the task of relation classification into a task of completing blanks, using a masking task for predicting possible relations when the language model is trained.

Since the language model is not so effective in terms of specific domains or specific tasks. Peters et al. [19] embed the contents of multiple knowledge bases into BERT, and the knowledge-enhanced BERT also achieves better results in downstream tasks such as relation extraction. Wang et al. [20] configured a neural adapter for each kind of injected knowledge in order not to let the injected historical knowledge be washed away, thus allowing the fusion of multiple knowledge bases and making the model have better results.

Recently, great progress has also been made in Few-Shot Relation Extraction. Qin et al. [21] used a continual few-shot relation learning method based on embedding space regularization and data augmentation to avoid catastrophic forgetting of previous tasks. Liu et al. [22] proposed a direct addition method that introduces relational information, generates a relational representation by joining two relations and then adds it to the model for training and prediction. Chia et al. [23] worked on the task setting of the zero-shot relation triplet extraction task. Unifying language model prompts and structured text methods, the relationship samples were generated by conditional processing of relations with structured prompts templates and decoded according to the triplet search decoding method.

Also, ensemble learning is effective in the field of relational extraction, i.e., the performance of relational extraction models can be improved by ensemble learning. Han et al. [24] used multiple semi-supervised learning methods to form a new semi-supervised learning method based on ensemble learning, which is well applied to relational classification. Kim et al. [25] used four classifiers, CRF, CRFext, SEARN, and Bi-LSTM, for relational classification, and finally learned the four models together in an ensemble-learning manner, which also achieved better results. Yang et al. [26] constructed a more efficient and robust relationship extractor based on a joint integrated neural network through the proposed adaptively enhanced multiple LSTM networks attention. Christopoulou et al. [27] used BiLSTM-CRF and feature-based CRF models as sub-models thus building ensemble-learning algorithms and using the integrated algorithms for extracting relationships between drugs and achieving better results. Rim et al. [28] propose a method to combine predictions from CNN and RNN into an integrated model to perform relational classification and extraction simultaneously, as well as a choice of weighted cross-entropy as the objective function and an up-sampling strategy to mitigate the negative effects of category imbalance.

Although the current sentence-level relation extraction methods have achieved great success, there is still much room for improvement: the feature information extraction for sentences is not sufficient, making the language model overly fine-tuned to downstream tasks, resulting in the language model not fully exploiting its performance; the utilization of feature vectors is not sufficient; the choice of loss functions is too obsolete.

Therefore, inadequate extraction of information about sentence features means that there is no information other than entity pair information as well as sentence information. If additional information could be added to identify the features of the relation extraction task so that more information would be encoded and injected into the model. It would also allow the language model to have more understanding of the relation extraction task, which may enable the language model to perform to its full potential. Second, underutilization of feature vectors refers to the fact that the feature vectors are used only once during the propagation of the neural network, for example, the R-BERT method proposed by Wu et al. [15] utilizes the sentence and entity pair information only once. If the feature vector can be used more times, it may have a better improvement on the relation extraction results. To overcome the above problems, this paper proposes a sentence-level relation extraction method based on adding prompt information and feature reuse.

3. A RELATION EXTRACTION MODEL BASED ON PROMPT INFORMATION AND FEATURE REUSE

The model is presented in three main areas: the encoding layer, the model details, and the model optimization. In the encoding layer, the details of how the logarithmic data is modified and the features of the encoded matrix-vector are presented. In terms of model details, this paper subdivides the model into three parts: input layer, feature acquisition layer, and feature reuse layer, as shown in Figure 1. Finally, for the optimization part of the model, this paper presents the way the loss function used is composed.

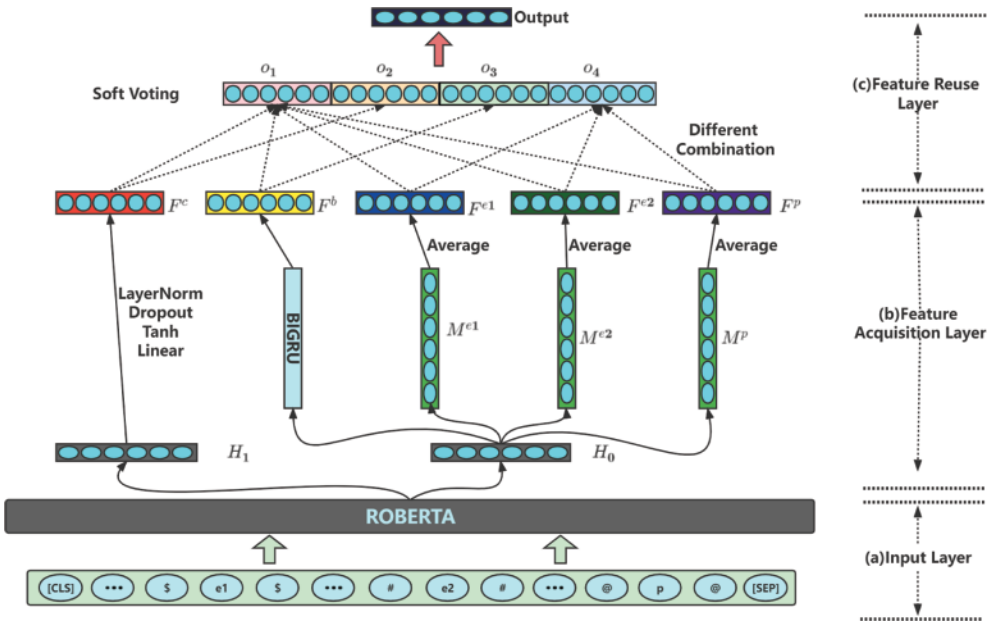

Figure 1. Overall model diagram, overall divided into three parts (a) Input Layer, (b) Feature Acquisition Layer, (c) Feature Reuse Layer.

3.1 Encoding Layer

For all the data in the dataset first perform a replacement operation, replacing "<e1>", "</e1>" in the dataset with the special tokens " $ ", "<e2>", "</e2>" is replaced by the special tokens "#", and finally at the end of the sentence add a prompt message " What is the relationship between entity one (e1) and entity two (e2) in the above sentence? ", and add the special tokens "@" before and after the prompt message. For example, a statement in the dataset " The <e1> legend </e1> was derived from a much older <e2> publication </e2>. " to " The $ legend $ was derived from a much older # publication #. @ What is the relationship between legend and publication in the above sentence? @ " as input.

In this paper, we use the pre-trained Roberta mode to encode the input sentences. For the input format specific to the Roberta model, we need to add "[*CLS*]" and "[*SEP*]" before and after the sentence to indicate the beginning and end of the sentence respectively. For the proposed model in this paper, five vector matrices are designed as inputs to the model. For modified statements S after Roberta encoding, all statements are set to a maximum length L, and statements of insufficient length are made up with zeros. The sentence S can then be represented as a set of word vectors $input_{ids}$ noted as $I^i = \{x_1^i, x_2^i, \ldots, x_L^i\}$. To perform the self-attention operation on the specified words, Roberta follows the matrix-vector $attention_{mask}$ proposed in Bert notated as $M^a = \{x_1^a, x_2^a, \ldots, x_L^a\}$. The matrix-vector M^a in which all positions are one, except for the complementary zero position, which is zero. The vector matrices of entity one, entity two, and prompt message are denoted as $M^{e1} = \left\{x_1^{e1}, x_2^{e1}, \ldots, x_L^{e1}\right\}$, $M^{e2} = \left\{x_1^{e2}, x_2^{e2}, \ldots, x_L^{e2}\right\}$, $M^p = \{x_1^p, x_2^p, \ldots, x_L^p\}$, the vector matrices M^{e1}, M^{e2}, M^p have zero values at all positions except for the position identified by the special symbol, which is one. That is the vector matrices for each of the five inputs are I^i, M^a, M^{e1}, M^{e2}, M^p.

3.2 Model Details

This section will be divided into three parts to introduce the detailed parts of the model. The three sections are Input Layer, Feature Acquisition Layer, and Feature Reuse Layer, which describe the model in detail in terms of input, feature acquisition, and model type optimization. The specific details are shown in Figure 1.

3.2.1 Input Layer

In the input layer of the model, the matrix-vector I^i and the matrix-vector M^a are first fed into the pre-trained Roberta mode, which outputs a hidden layer H. Then the matrix-vector M^{e1}, the matrix-vector M^{e2}, and the matrix-vector M^p are input into the model to be multiplied by the hidden layer H. From this, information can be extracted about the entity pair with the whole sentence and the prompt part with the whole sentence.

3.2.2 Feature Acquisition Layer

The hidden layer H is used as the output of Roberta, and the feature vector H_0 contains the information of the whole sentence. Denote H_1 as the feature vector of sentence information $F^c = \left\{y_1^c, y_2^c, \ldots, y_{L_H}^c\right\}$, the

L_H represents the length of the output of the hidden layer H. Then put the feature vector H_0 as the input of BIGRU, so that the information of the whole sentence can be extracted by BIGRU again, thus strengthening the features of the input. For each element in the BIGRU input sequence, the following function is computed for each layer:

$$r_t = \sigma W_r[h_{t-1}, H_0] \tag{1}$$

$$z_t = \sigma W_z[h_{t-1}, H_0] \tag{2}$$

$$h_t' = \tanh(W_{h'}[r_t \odot h_{t-1}, H_0]) \tag{3}$$

$$h_t = z_t \odot h_t + (1 - z_t) \odot h_t' \tag{4}$$

where σ is the sigmoid activation function, $\odot$ is the product of terms, W_r, W_z, and W_h are the parameters of the GRU network. The h_t is considered as the output of BIGRU, and h_t is denoted as the feature vector $F^b = \left\{y_1^b, y_2^b, \ldots, y_{L_H}^b\right\}$ extracted by BIGRU.

As shown in Algorithm 1, take the matrix-vector M^{e1}, the matrix-vector M^{e2}, matrix-vector M^p respectively, and H_0 multiplying by each other, assuming that H_0 the vectors are represented as $\{\beta_1, \ldots, \beta_{L_H}\}$ with the following equation.

Algorithm 1. Figure out the average vector.

Input: Roberta's hidden layer vector, H_0, masked vector Mask-Tensor, M_n;
Output: The average of the masked vector, Avg_n;
1: **function** AVG (H_0, M_n)
2: Add one dimension to the middle of a two dimensional vector M_n to make a three dimensional vector;
3: Figure out the length L of the nonzero number in vector M_n;
4: Sum the matrix vector M_n and the matrix vector H_0, *Sum*;
5: Avg_n = Sum / L;
6: **return** Avg_n;
7: **end function**

$$M_{sum} = \sum_{i=1}^{L_H} x_i \beta_i = x_1 \beta_1^T + x_2 \beta_2 + \ldots + x_{L_H} \beta_{L_H} \tag{5}$$

Taking the calculated matrix-vector M_{sum} and dividing it by the length of the masked part of each matrix-vector to obtain the final average vector, the resulting result is denoted as $F^{e1} = \left\{y_1^{e1}, y_2^{e1}, \ldots, y_{L_H}^{e1}\right\}$, $F^{e2} = \left\{y_1^{e2}, y_2^{e2}, \ldots, y_{L_H}^{e2}\right\}$, $F^p = \left\{y_1^p, y_2^p, \ldots, y_{L_H}^p\right\}$. That is, five eigenvectors are obtained in the feature acquisition layer F^c, F^b, F^{e1}, F^{e2}, F^p as the input to the next step.

3.2.3 Feature Reuse Layer

Five features F^c, F^b, F^{e1}, F^{e2}, F^p are taken as input in the feature reuse layer. As shown in Algorithm 2. The features are processed using Algorithm 2, first by regularizing the input feature vectors; Then, in order not

to overfit the model, it goes through the dropout layer, dropping some random neurons. Finally, after passing the function Tanh, the linear model is used to make the reduced dimensional output. The output obtained from each feature is then stitched together to obtain O_1. Putting F^c and F^b then go through the same operation separately, and the output from the linear layer to get O_2 and O_3. Finally, put F^{e1}, F^{e2}, F^p are also passed through Algorithm 2 to obtain O_4. Where the specific linear operation formula is as follows:

Algorithm 2. Figure out the hidden layer output.

Input: $Feature^c$, $Feature^b$, $Feature^{e1}$, $Feature^{e2}$, $Feature^p$ as a characteristic innput, *Features*;
Output: Output containing all characteristics, *Out*;
1: **function** FIGURE-OUT (Feature)
2: Create an output array Out_i
3: **for** choose $Feature_i$ in *Features* **do**
4: $Feature_i$ = LayerNorm ($Feature_i$);
5: $Feature_i$ = Dropout ($Feature_i$);
6: $Feature_i$ = Tanh ($Feature_i$);
7: $Feature_i$ = Linear ($Feature_i$);
8: **end for**
9: Add multiple $Feature_i$ to the array Out_i;
10: Normalize list Out_i to form the final output, *Out*;
11: **return** *Out*;
12: **end function**

$$O_i = w_i \tanh\left(F^i\right) + b_i \tag{6}$$

The obtained outputs O_1, O_2, O_3 and O_4 are then passed through SoftMax to obtain O_1', O_2', O_3' and O_4' the specific formula is as follows:

$$P\left(y|x\right) = \frac{e^{h(x,y_i)}}{\sum_{j=1}^{n} e^{h(x,y_i)}} \tag{7}$$

Finally, using the idea of ensemble-learning soft voting, the probabilities of each classification outcome in each output are summed and averaged to find the final relation classification output.

$$\text{output} = \frac{1}{4}\left(O_1' + O_2' + O_3' + O_4'\right) \tag{8}$$

3.3 Model Optimization

Since deep neural networks are prone to overfitting, regularization methods such as dropout are usually used to reduce the generalization error of the model during the training process. The dropout removes a random portion of units in each layer of the neural network to avoid overfitting the model. It is due to the randomness of dropout that Liang et al. [29] proposed a dropout-based loss function.

In this paper, we add cross-entropy to the loss function based on the above approach. The final loss function consists of cross-entropy, KL divergence, and negative log-likelihood loss. First, let each batch of

data pass through the forward neural network twice, before and after, and two different distributions can be obtained from Figure 2, respectively P1 and P2. Due to the randomness of dropout, the forward pass is also slightly different in spite of passing the same model twice. P1 left path is dropped with the output distribution and P2 right path is dropped with the output distribution is not the same. For this purpose the KL divergence is used to describe the difference between two distributions noted as L_{kl}^{i} as follows:

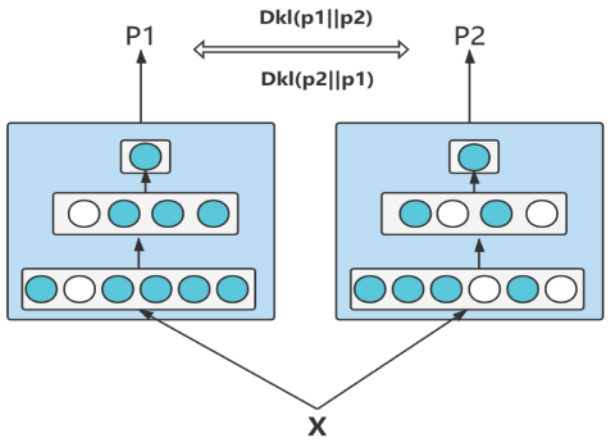

Figure 2. Dropout specific process.

$$L_{kl}^{i}=\frac{1}{2}\left(D_{kl}P_{1}^{i}\left(y_{i}|x_{i}\right)||P_{2}^{i}\left(y_{i}|x_{i}\right)+D_{kl}P_{2}^{i}\left(y_{i}|x_{i}\right)||P_{1}^{i}\left(y_{i}|x_{i}\right)\right) \tag{9}$$

Then the cross-entropy L_{CE}^{i} and the negative log-likelihood loss L_{NLl}^{i} are used to find a difference value of the two results respectively.

$$L_{CE}^{i}=-P_{1}^{i}\left(y_{i}|x_{i}\right)logP_{1}^{i}\left(y_{i}|x_{i}\right)-P_{2}^{i}\left(y_{i}|x_{i}\right)logP_{2}^{i}\left(y_{i}|x_{i}\right) \tag{10}$$

$$L_{NLl}^{i}=-logP_{1}^{i}\left(y_{i}|x_{i}\right)-logP_{2}^{i}\left(y_{i}|x_{i}\right) \tag{11}$$

Finally, the losses L_{kl}^{i}, L_{CE}^{i}, and L_{NLl}^{i} are summed to obtain a final loss function L_{loss}^{i}.

$$L_{loss}^{i}=(1-\alpha)(L_{CE}^{i}+L_{NLL}^{i})+\alpha L_{kl}^{i} \tag{12}$$

4. EXPERIMENTS AND ANALYSIS

The experiments attempt to demonstrate the enhancement of prompt information, feature reuse, and loss functions on the performance of the model, thus further enhancing the effectiveness of existing relation classification methods. The dataset is first presented, then the model in this paper is compared with existing methods, and finally, the impact of each part of the model on the model results is explored.

4.1 Dataset

For the data part, the dataset used in this paper is the SemEval-2010 task 8 relational dataset. The dataset contains 10717 samples, 8000 samples for training, and 2717 samples for testing. The dataset contains 9 semantic relationship types and 1 other relationship type Other, the relationships are ordered. The directionality of the relations effectively doubles the number of relations, since entity pairs are considered to be correctly labeled only if the order is also correct. Cause-Effect (e1, e2) is different from Cause-Effect (e2, e1). So ultimately 19 relationships exist, for the relationships contained in the dataset and the number of individual relationships as shown specifically in Table 2.

Table 2. Specific Number Of Data Types In The Dataset.

Relation	Train	Test
Cause-Effect	1003	328
Instrument-Agency	504	156
Product-Producer	717	231
Content-Container	540	192
Entity-Origin	716	258
Entity-Destination	845	292
Component-Whole	941	312
Member-Collection	690	233
Message-Topic	634	261
Other	1410	454
Totle	8000	2717

4.2 Parameter Setting

In this paper, we use the grid search algorithm to adjust the optimal parameters, the maximum sentence length L $\in$ {120, 150, 200, 250, 300}, the size of each batch of data BATCH-SIZE $\in$ {4, 8, 16}, the total number of training EPOCHS $\in$ {8, 10, 12, 14, 16}, the neural network dropout DROPOUT-RATE $\in$ {0.1, 0.2, 0.3, 0.4, 0.5}, the LEARNING-RATE $\in$ {1e–5, 2e–5, 3e–5, 4e–5, 5e–6}, loss function KL scatter percentage ratio KL-RATE $\in$ {0.4, 0.5, 0.6, 0.7, 0.8}, hidden layer length H-L $\in$ {100, 150, 200, 250, 300}. The optimal configuration of parameters is obtained as L=150, BATCH-SIZE=10, EPOCHS=12, DROPOUT-RATE=0.1, LEARNING-RATE= 1e-5, KL-RATE=0.7, H-L=200.

4.3 Comparison of Different Methods

The proposed model, denoted as RPR, is compared with the previous methods TRE [12], Entity-Aware BERT [14], R-BERT [15], PTR [18], and Skeleton-Aware BERT [17]. The specific results are shown in Table 3.

Table 3. Different models for relation extraction.

Model	F1-score
TRE	87.1
Entity-Aware BERT	89.0
R-BERT	89.25
PTR	89.9
Skeleton-Aware BERT	90.36
RPR	90.70

(1) Comparison with TRE [12] method. The TRE approach learns implicit linguistic features from a plain text corpus and combines them in a self-attention Transformer architecture. It does not take into account information other than entity pairs and sentence-level information. Whereas, the RPR method adds prompt information to be able to better extract features about the relation extraction task.

(2) Comparison with Entity-Aware BERT [14] comparison of the methods. The Entity-Aware BERT method can accomplish the multi-entity relation extraction task by encoding only once. However, it does not take into account the utilization of feature information. The RPR method, on the other hand, reuses entity pairs as well as sentence-level information multiple times, effectively alleviating the problem of the underutilization of feature vectors.

(3) Comparison with R-BERT [15] method. The R-BERT method uses a pre-trained BERT language model and merges information from the target to handle the relation classification task. But it does not sufficiently extract information from the target. The RPR approach, with the addition of a prompt to emphasize the target information, allows the language model to be fully understood.

(4) Comparison with PTR [18] comparison of the methods. The PTR approach proposes prompt-based learning by adding a piece of information other than entity pairs, sentence-level information, and applying the mask training task of the language model to predict the classification, but it does not take into account that the predicted categories are too exotic, which leads to unsatisfactory results. The RPR method, with the addition of information, still follows the idea of the classification task and is able to better infer the classification and achieve better results.

(5) Comparison with Skeleton-Aware BERT [17] comparison of methods. The Skeleton-Aware BERT method extracts syntactic indicators guided by syntactic knowledge and merges syntactic indicators and whole sentences into a better relational representation. But it does not take into account the performance of the language model and the degree of feature utilization. The RPR method uses a better ROBERTA language model, as well as feature reuse of the components of each part of the model, which improves the accuracy and F1-score values.

4.4 Effect of Model Components on the Model

This paper has demonstrated strong empirical results based on the proposed method, and to further understand the specific contribution of each component of the proposed method, the following control

group experiment was set up for this purpose. For the pre-trained language models, BERT and ROBERTA were used as the base models to set up control trials, respectively. Two types of inputs are used in this paper, one using the original input to mark special symbols for only two entities in the sentence, and the other input using a prompt-based input, a prompt message is added at the end of the sentence. For the specific models, three groups are also used, the first group is based on the pre-trained language model for classification, the second group adds BILSTM on top of the pre-trained language model, and the third group adds BIGRU on top of the pre-trained language model. For the loss functions, two groups are used in this paper, one just using the cross-entropy loss function and the other using the loss function proposed in this paper. All experiments were performed using grid tuning reference to obtain the final results under the optimal parameters.

In Figure 3 (a), (b) it can be seen that the overall model reaches its maximum value with about 10 Epochs of fine-tuning and stabilizing. It can also be seen that the model model-roberta-prompt-gru achieves the maximum value of all models. And the overall performance of the model is also improved after using the improved loss function in this paper compared with the previous model using only the cross-entropy loss function.

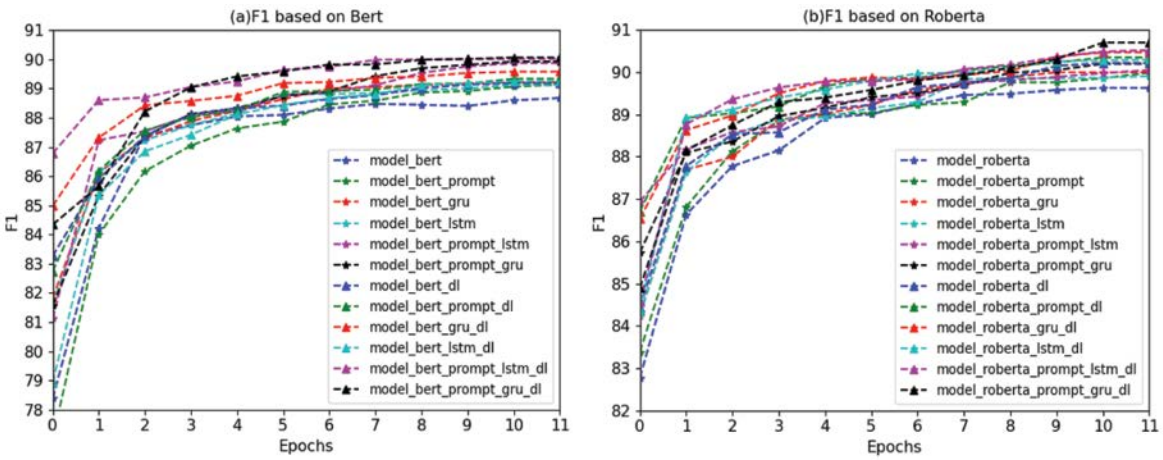

Figure 3. Results of the specific effects of each component of the model on the model.

From Table 4, and Table 5, it can be seen that a large number of experiments were done in this paper to verify the conclusions. Where dl represents the loss function used in this paper indicates. The overall performance of the Roberta model is better than that of Bert, and the effectiveness of the loss function proposed in this paper can also be seen in the table. And also the maximum value of 90.70 is obtained in Roberta-bigru-prompt-dl.

Table 4. Experimental comparison based on the BERT.

Model	F1-score	Model	F1-score
bert	88.68	bert-dl	89.19
bert-prompt	89.14	bert-prompt-dl	89.34
bert-bilstm	89.22	bert-bilstm-dl	89.26
bert-bigru	89.14	bert-bigru-dl	89.58
bert-bilstm-prompt	89.88	bert-bilstm-prompt-dl	90.02
bert-bigru-prompt	89.93	bert-bigru-prompt-dl	90.07

Table 5. Experimental comparison based on the ROBERTA.

Model	F1-score	Model	F1-score
roberta	89.63	roberta-dl	90.22
roberta-prompt	89.99	roberta-prompt-dl	90.35
roberta-bilstm	90.00	roberta-bilstm-dl	90.30
roberta-bigru	89.91	roberta-bigru-dl	90.47
roberta-bilstm-prompt	90.05	roberta-bilstm-prompt-dl	90.52
roberta-bigru-prompt	90.20	roberta-bigru-prompt-dl	90.70

5. CONCLUSION AND FUTURE WORK

In this paper, an approach to sentence-level relation extraction based on adding prompt information and feature reuse is proposed. By adding a prompt message, the sentence is made more informative and allows the pre-trained ROBERTA mode to better understand the relation extraction task. On this basis, certain feature information is also reused in the model to constitute multiple output results, and the idea of integrated learning is used to soft-vote the output results, which enhances the robustness of the experimental results. Finally, the model is optimized by using cross-entropy, KL divergence, and the sum of negative log-likelihood losses as loss functions, and better results are achieved on the SemEval-2010 task 8 relational dataset. This also enables more accurate identification of the relationships between entities in the knowledge graph building blocks in various fields such as medicine, movie, and music, and provides a reliable guarantee for the accuracy of the knowledge graph construction. The direction of future work is to be able to introduce graph neural networks while employing prompt information and feature reuse, which can better capture the information of sentence and entity pairs.

AUTHOR CONTRIBUTIONS

Xin Zhang was responsible for experimental idea construction, method design, data analysis and thesis writing. Ping Feng was responsible for the thesis review and revision, experimental investigation, and experimental supervision. Yingying Wang, Jian Zhao and Biao Huang are responsible for project management and experimental hardware preparation.

ACKNOWLEDGEMENTS

This work is supported by the project of the Ministry of Education (research on the construction and application of quantum encryption cloud service system based on big data analysis of web content (2019JB328L06)) and the scientific research planning project of the Jilin Provincial Education Department (construction and application of medical knowledge graph for chronic diseases of the elderly (JJKH20210614KJ)).

REFERENCES

[1] Devlin, J., et al.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. North American Chapter of the Association for Computational Linguistics (1), 4171–4186 (2019)
[2] Liu, Y., et al.: RoBERTa: A Robustly Optimized BERT Pretraining Approach. Computing Research Repository (2019)
[3] Radford, A., et al.: Improving language understanding by generative pre-training. OpenAI (2018)
[4] Zeng, D., et al.: Relation Classification via Convolutional Deep Neural Network. International Conference on Computational Linguistics, 2335–2344 (2014)
[5] Zhang, S., et al.: Bidirectional Long Short-Term Memory Networks for Relation Classification. Pacific Asia Conference on Language, Information and Computation (2015)
[6] Zhang, Y., Qi, P., Manning, C.D.: Graph Convolution over Pruned Dependency Trees Improves Relation Extraction. Conference on Empirical Methods in Natural Language Processing, 2205–2215 (2018)
[7] Shen, Y., Huang, X.J.: Attention-Based Convolutional Neural Network for Semantic Relation Extraction. International Conference on Computational Linguistics, 2526–2536 (2016)
[8] Zhou, P., et al.: Attention-Based Bidirectional Long Short-Term Memory Networks for Relation Classification. Annual Meeting of the Association for Computational Linguistics (2) (2016)
[9] Guo, Z., Zhang, Y., Lu, W.: Attention Guided Graph Convolutional Networks for Relation Extraction. Annual Meeting of the Association for Computational Linguistics (1), 241–251 (2019)
[10] Lee, J., Seo, S., Choi, Y.S.: Semantic Relation Classification via Bidirectional LSTM Networks with Entity-Aware Attention Using Latent Entity Typing. Symmetry 11(6), 785 (2019)
[11] Peters, M.E., et al.: Deep Contextualized Word Representations. North American Chapter of the Association for Computational Linguistics, 2227-2237 (2018)
[12] Alt, C., Hübner, M., Hennig, L.: Improving Relation Extraction by Pre-trained Language Representations. Conference on Automated Knowledge Base Construction (2019)
[13] Vaswani, A., et al.: Attention is All you Need. Conference on Neural Information Processing Systems, 5998–6008 (2017)
[14] Wang, H., et al.: Extracting Multiple-Relations in One-Pass with Pre-Trained Transformers. Annual Meeting of the Association for Computational Linguistics (1), 1371–1377 (2019)
[15] Wu, S., He, Y.: Enriching Pre-trained Language Model with Entity Information for Relation Classification. International Conference on Information and Knowledge Management, 2361–2364(2019)
[16] Tian, Y., et al.: Dependency-driven Relation Extraction with Attentive Graph Convolutional Networks. Annual Meeting of the Association for Computational Linguistics (1), 4458–4471 (2021)
[17] Tao, Q., et al.: Enhancing Relation Extraction Using Syntactic Indicators and Sentential Contexts. IEEE International Conference on Tools with Artificial Intelligence, 1574–1580 (2019)
[18] Han, X., et al.: PTR: Prompt Tuning with Rules for Text Classification. AI Open 3, 182–192 (2022)

[19] Peters, M.E., et al.: Knowledge Enhanced Contextual Word Representations. Conference on Empirical Methods in Natural Language Processing (1), 43–54 (2019)
[20] Wang, R., et al.: K-Adapter: Infusing Knowledge into Pre-Trained Models with Adapters. Annual Meeting of the Association for Computational Linguistics, 1405–1418 (2021)
[21] Qin, C., Joty, S.: Continual Few-shot Relation Learning via Embedding Space Regularization and Data Augmentation. Annual Meeting of the Association for Computational Linguistics (1): 2776–2789 (2022)
[22] Liu, Y., et al.: A Simple yet Effective Relation Information Guided Approach for Few-Shot Relation Extraction. Annual Meeting of the Association for Computational Linguistics, 757–763 (2022)
[23] Chia, Y.K., et al.: RelationPrompt: Leveraging Prompts to Generate Synthetic Data for Zero-Shot Relation Triplet Extraction. Annual Meeting of the Association for Computational Linguistics, 45–57 (2022)
[24] Han, Z., Yin, S.: Research on semi-supervised classification with an ensemble strategy. International Conference on Sensors, Mechatronics and Automation, 681–684 (2016)
[25] Kim, Y., Meystre, S.M.: Ensemble method-based extraction of medication and related information from clinical texts. Journal of the American Medical Informatics Association 27(1), 31–38 (2020)
[26] Yang, D., Wang, S., Li, Z.: Ensemble Neural Relation Extraction with Adaptive Boosting. International Joint Conference on Artificial Intelligence, 4532–4538 (2018)
[27] Christopoulou, F. et al.: Adverse drug events and medication relation extraction in electronic health records with ensemble deep learning methods. Journal of the American Medical Informatics Association 27(1), 39–46 (2020)
[28] Rim, K. et al.: Reproducing Neural Ensemble Classifier for Semantic Relation Extraction inScientific Papers. International Conference on Language Resources and Evaluation, 5569–5578 (2020)
[29] Liang, X., et al.: R-Drop: Regularized Dropout for Neural Networks. Conference on Neural Information Processing Systems, 10890–10905 (2021)

AUTHOR BIOGRAPHY

Ping Feng is an associate professor and master supervisor at Changchun University. She is currently pursuing her PhD at the College of Computer Science and Technology, Jilin University. Her current research interests include knowledge graph embedding and knowledge acquisition.
ORCID: 0000-0003-4865-1454

Xin Zhang is a graduate student in the College of Cyber Security, Changchun University. His current research interests include knowledge graphs, relation extraction, and link prediction.
ORCID: 0000-0002-7055-3650

Jian Zhao is a professor and master supervisor advisor at Changchun University. His current research interest is in cyber security.
ORCID: 0000-0003-3265-6461

Yingying Wang is a graduate student in the College of Cyber Security, Changchun University. Her current research interests include Knowledge graph, machine learning, cybersecurity.
ORCID: 0000-0002-7150-1908

Biao Huang is a graduate student in the College of Cyber Security, Changchun University. His current research interests include image captioning, machine learning, and cybersecunty.
ORCID: 0000-0002-2957-904X