mathematics

MDPI

Article

A Concise Relation Extraction Method Based on the Fusion of Sequential and Structural Features Using ERNIE

Yu Wang [1], Yuan Wang [2], Zhenwan Peng [1], Feifan Zhang [1] and Fei Yang [1,*]

[1] School of Biomedical Engineering, Anhui Medical University, Hefei 230001, China; wangyu@ahmu.edu.cn (Y.W.); pengzhenwan@ahmu.edu.cn (Z.P.); ffz@ahmu.edu.cn (F.Z.)
[2] Hefei Institutes of Physical Science, Chinese Academy of Sciences, Hefei 230001, China; wangyuan@iim.ac.cn
* Correspondence: yangfei@ahmu.edu.cn

Abstract: Relation extraction, a fundamental task in natural language processing, aims to extract entity triples from unstructured data. These triples can then be used to build a knowledge graph. Recently, pre-training models that have learned prior semantic and syntactic knowledge, such as BERT and ERNIE, have enhanced the performance of relation extraction tasks. However, previous research has mainly focused on sequential or structural data alone, such as the shortest dependency path, ignoring the fact that fusing sequential and structural features may improve the classification performance. This study proposes a concise approach using the fused features for the relation extraction task. Firstly, for the sequential data, we verify in detail which of the generated representations can effectively improve the performance. Secondly, inspired by the pre-training task of next-sentence prediction, we propose a concise relation extraction approach based on the fusion of sequential and structural features using the pre-training model ERNIE. The experiments were conducted on the SemEval 2010 Task 8 dataset and the results show that the proposed method can improve the *F1* value to 0.902.

Keywords: relation extraction; pre-training models; BERT; ERNIE; shortest dependency path; fusion methods

MSC: 68T50; 68T07

Citation: Wang, Y.; Wang, Y.; Peng, Z.; Zhang, F.; Yang, F. A Concise Relation Extraction Method Based on the Fusion of Sequential and Structural Features Using ERNIE. *Mathematics* **2023**, *11*, 1439. https://doi.org/10.3390/math11061439

Academic Editors: Nebojsa Bacanin and Catalin Stoean

Received: 3 February 2023
Revised: 11 March 2023
Accepted: 14 March 2023
Published: 16 March 2023

1. Introduction

Electronic books, documents, and other forms of literature contain a wealth of knowledge in specific fields, such as finance, medicine, and agriculture. However, the unstructured nature of this knowledge makes it difficult to apply automatic deduction and reasoning. Therefore, extracting the structured entity triples from the unstructured textual sources has significant research value and economic benefits [1,2]. As shown in Figure 1, "A common side effect of glinides is hypoglycemia." is selected from the "Guideline for the prevention and treatment of type 2 diabetes mellitus in China (2020 edition)", and the corresponding entity triple "<*Glinides, Hypoglycemia, Side-effect*>" can be extracted using the knowledge extraction approach. Furthermore, the knowledge graph (KG) can be constructed by aggregating these entity triples. The most common knowledge extraction approach for unstructured data consists of two types of basic natural language processing (NLP) task: named entity recognition (NER) and relation extraction (RE). NER aims to recognize the named entities in a sentence, such as the "Glinides" and "Hypoglycemia" in Figure 1, which can be considered as the nodes in the KG. However, the named entities recognized by the NER are not related to each other. The relations between entities need to be defined by the RE. For example, in Figure 1, the relation 'Side-effect' between 'Glinides' and 'Hypoglycemia' can be considered as an edge between nodes in the knowledge graph. Finally, a primary KG can be constructed through the above NLP tasks. Therefore, since the RE tasks play a vital role in the process of knowledge extraction, it is essential to study and design a more efficient and concise RE method to improve the accuracy of the KG

and enhance its performance in relevant application scenarios, such as recommendation systems [3].

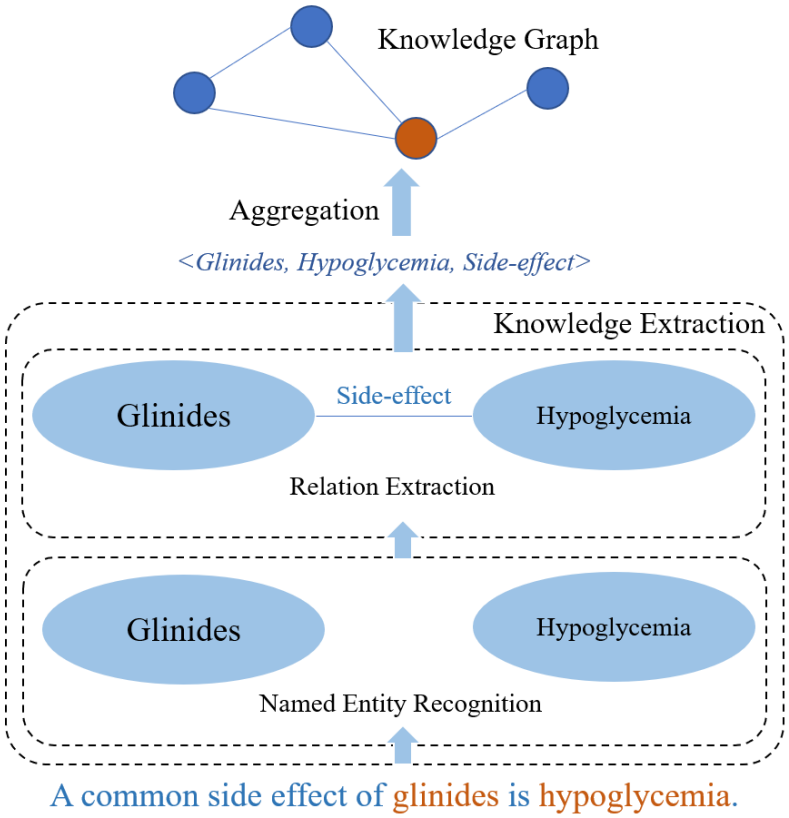

Figure 1. The standard knowledge extraction method adapted for unstructured data.

As mentioned above, relation extraction, as a basic NLP task, aims to define the relations between entities from unstructured textual sources. On the one hand, RE tasks can be classified as sentence-level or document-level tasks according to the distribution of entities [4,5]; on the other hand, depending on the number of relations between two entities, RE tasks can also be divided into binary or n-ary classification tasks [6,7]. In this study, we focus on binary RE tasks, where the two entities are located in a single sentence because such tasks are the basis of document-level or n-ary RE tasks. For binary RE tasks, previous research has found that the performance of machine learning [8] and statistical approaches [9] is highly dependent on the quality of feature engineering. In contrast, deep learning models, such as the convolutional neural network (CNN) [10] or the recurrent neural network (RNN) [11], can provide improved performance without additional manual feature selection. However, regardless of the approach used, the essence is still to generate representations with abstract semantics and predict the categories of relations based on them. Considering that such models can only capture the semantic knowledge within training sets, they are unsatisfactory because the degree of improvement is still limited by the accessibility of the labeled training sets. Therefore, how to enrich the semantic knowledge contained in the representations is the key to improving the performance of RE tasks.

Recently, pre-training models, such as the BERT [12] and ERNIE [13], have received considerable attention and have enhanced the performance of various NLP tasks, including RE tasks. Pre-training models obtain the prior semantic and syntactic knowledge from the unlabeled corpus through the pre-training process. Therefore, distinct from previous approaches, the methods based on pre-training models can transfer prior knowledge to downstream tasks through fine-tuning, which can be considered as a transfer learning pro-

cess. Recent evidence suggests that the pre-training models can achieve better performance based on an identical annotated training set [14]. However, there are still some relevant areas of research that need to be focused on.

Firstly, most studies in the field of RE have focused only on sequential or structural features alone. For example, quite a few studies construct their models based on raw text sequences, where the generated representations contain sequential features. However, there are other researchers who input structured data (such as the shortest dependency path (SDP)) into the model and obtain the structured features. Therefore, the effect of fusing sequential and structural features on the performance of RE tasks is still unclear. Secondly, there has been no detailed investigation of which representations generated by pre-training models are more appropriate for RE tasks. Since the RE task can be viewed as an N-to-1 sequence modeling process, researchers use the feature of the `[CLS]` token for classification [14,15], similar to a text classification task [12]. However, different from a text classification task, an RE task should pay more attention to the entity pairs in the input sequence. Therefore, a method is needed to more conveniently introduce the relevant information of entity pairs into pre-training models and improve the effectiveness of the RE tasks.

The importance and originality of this study are that it explores how to use the fused features effectively and concisely for the relation extraction task. The main contributions of this paper can be summarized as follows:

1. Firstly, for the sequential feature, we verify in detail which types of representations generated by pre-training models can effectively improve performance. Since the RE task can be considered as a text classification task, previous approaches used the representation of `[CLS]` (i.e., the sentence-level feature) to obtain the relation label, as in the work reported by Wang et al. [14] and Han et al. [16]. However, the RE task will focus more on the information of entity pairs in the input sequence. To this end, we extend the input and output sides of the pre-training model and search for the representations that can improve the performance of the RE task. Specifically, for the input side of the pre-training models, we insert the entity location tokens `[ES]` and `[EE]` into the raw text and try to add the entity embeddings to the input embeddings; for the output side, we experimented with the performance when applying the representations of the `[CLS]` and `[ES]` on the RE task.
2. Secondly, we propose a concise RE approach based on the fusion of sequential and structural information using a pre-training model. Previous work has attempted to extract the relation label leveraging structural features, such as the BERT-DP proposed by Cho et al. [17]. However, these methods require training a graph convolutional network (GCN) to extract structural features, which may introduce new errors and make the model even more complex. Inspired by the pre-training task of next sentence prediction (NSP) [12], we use the `[SEP]` token to concatenate the shortest dependent paths of entity pairs with the original sequence and then obtain the representation of the `[CLS]` token which can be regarded as the fusion of sequential and structural features. In addition to BERT, we also test the performance of the above approach on ERNIE, which incrementally constructs pre-training tasks and then extracts the lexical, syntactic, and semantic information of these constructed tasks via continual multi-task learning. It is also the first study to extract entity relations for leverage in the NSP task.
3. Thirdly, the experiments are conducted on the SemEval 2010 Task 8 dataset. The experimental work presented here shows that, compared with previous studies, the proposed model is not only more convenient but can also increase the *F1* value to 0.902.

The overall structure of this study consists of six sections. A brief overview of the related work is given in section two. The third section presents the methodology used for this study. The experimental results are presented in section four, while the discussion is presented in section five. Finally, section six concludes this study with a summary.

2. Related Work

As shown in Table 1, the common relation extraction methods include those based on machine learning, deep learning and pre-training models.

Table 1. Related work in the field of RE.

Method Categories	Authors	Comments
Machine learning	Rink et al. [18] Zhao et al. [19] Kim et al. [20] Bui et al. [8]	The key to machine learning methods lies in manual feature selection, which is time-consuming and labor-intensive.
Deep learning	Rink et al. [18] Zeng et al. [10] Choi [21] Liu et al. [22] Peng et al. [23] Zhou et al. [24] Xu et al. [25] Li et al. [26] Wang et al. [27] Xu et al. [28] Corbett and Boyle [29]	The degree of improvement is still limited by the fact that the labeled training sets are inaccessible.
Pre-training models	Wang et al. [14] Han et al. [16] Cho et al. [17] Chauhan et al. [30]	Ignores the possibility of improving the performance of the RE task by fusing the sequential and structural features using pre-training models.

2.1. Machine Learning Methods

Machine learning methods for RE tasks typically include the support vector machine (SVM) and K-nearest neighbor (KNN) methods. These methods map manually selected syntactic or semantic features into a high-dimensional space using kernel functions to make them linearly separable. For example, Rink et al. [18] performed the classification using SVM classifiers and several features that capture the context, semantic role affiliation, and possible pre-existing relations of the nominals. Zhao et al. [19] combined cues from different levels of syntactic processing using kernel methods and tested different kernels on the SVM and KNN. Kim et al. [20] proposed an efficient and scalable system using a linear kernel to identify information on drug–drug interactions. Bui et al. [8] proposed a feature-based approach to extract drug–drug interactions from the text. They mapped each candidate drug–drug interaction pair from a dataset into an appropriate syntactic structure to generate feature vectors, which were then used to train an SVM classifier. It can be seen that the key to machine learning methods lies in manual feature selection, which is time-consuming and labor-intensive.

2.2. Deep Learning Methods

Deep learning has become popular with the improvement of computer hardware, especially graphics processing units (GPUs). The most common deep learning methods are convolutional neural networks (CNNs), which can capture local features, and recurrent neural networks (RNNs), which are better at modeling sequential data [31,32]. Previous research has shown that the performance of medical text classification can be enhanced through CNNs or RNNs.

2.2.1. Convolutional Neural Networks

The CNNs were originally developed for image recognition and later extended to NLP, speech processing, and other fields [31]. The RE tasks based on these models typically use convolutional kernels with pooling operations to obtain the features of sequences and then

predict the relations. For example, Zeng et al. [10] exploited a CNN to extract sentence-level features that were fed into a softmax classifier to predict the relationship between two tagged nouns. Choi [21] and Liu et al. [22] proposed a CNN-based method for the extraction of protein–protein interactions and drug–drug interactions, respectively. Peng et al. [23] proposed a multichannel dependency-based CNN model that applies one channel to the embedding vector of each word in the sentence, and another channel to the embedding vector of the head of the corresponding word. Zhou et al. [24] performed convolution operations on the SDP to produce representations that contain the deep semantics of dependency directions and dependency relation tags.

2.2.2. Recurrent Neural Networks

The RNNs, unlike CNNs, were originally designed to be used for sequence modeling and can capture temporal features, which is crucial for the performance of the RE task when the distance between target entities is large [33]. For example, Xu et al. [25] utilized deep RNNs for relation classification, and they also proposed a data augmentation method by exploiting the directionality of relations. A variant of RNN, long short-term memory (LSTM), which was designed to address the gradient vanishing and exploding problems in RNNs by introducing a gate mechanism and memory cell, is more popularly applied [34]. For instance, Li et al. [26] proposed a joint model to extract biomedical entities and their relations simultaneously. The relation extraction part of this model consists of a bi-directional long short-term memory (BiLSTM) layer and a softmax layer. Wang et al. [27] proposed a deep neural network model for the extraction of drug–drug interactions by introducing the dependency-based technique into a BiLSTM layer. Xu et al. [28] proposed a new BiLSTM-based method that combines biomedical resources with lexical information and entity position information to extract drug–drug interactions from the biomedical literature. Corbett and Boyle [29] presented a system using multiple LSTM layers to analyze candidate chemical–protein interactions.

As can be seen, the core of these deep learning approaches is to use CNNs or RNNs to generate the features of either raw text sequences or SDPs. Some approaches further combine these features with entity features, lexical features, and external knowledge to predict the relations between entity pairs. However, the representations generated by the deep learning approaches can only gain semantic knowledge from the training sets. The degree of improvement is still limited by the fact that the labeled training sets are inaccessible.

2.3. Pre-Training Models

Inspired by ImageNet in the computer version field, researchers have proposed pre-training models such as BERT and ERNIE for NLP tasks. The typical structures of pre-training models is shown in Figure 2. The representations of the input tokens are generated by the multi-layer transformer blocks with self-attention heads [35]. These models obtain prior semantic knowledge from the unlabeled corpora through different pre-training tasks, which can be considered as self-learning tasks. They outperform deep learning models on various NLP tasks. As shown in Table 2 (Considering that the pre-training models are not the focus of this paper, we only validate the proposed relation extraction method based on BERT and ERNIE, but not other pre-training models such as RoBERTa and XLNet.), the structure of BERT and ERNIE is the same. The main differences between them are the pre-training tasks and the pre-training corpora. In short, both BERT and ERNIE perform the pre-training task of next sentence prediction (NSP), but ERNIE further goes on to incrementally build pre-training tasks and then learn lexical, syntactic, and semantic information on these built tasks via continuous multi-task learning. In addition, the number of parameters varies between the different models in the ERNIE series. The base model contains 12 layers, 12 self-attention heads, and a 768-dimension hidden size, while the large one contains 24 layers, 16 self-attention heads, and a 1024-dimension hidden size.

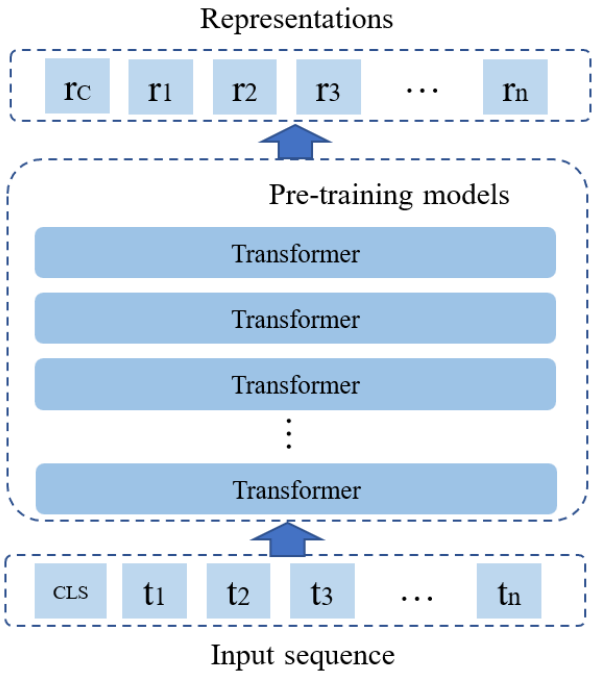

Figure 2. The typical structure of a pre-training model in the field of NLP.

Since the RE task can be considered as an N-to-1 sequence modeling process, researchers have performed classification through the representation of the `[CLS]` token [14]. However, the previous approaches ignored the possibility of improving the performance of the RE task by fusing the sequential and structural features using pre-training models.

Table 2. Parameters, pre-training tasks, and corpora of BERT and ERNIE.

Pre-Training Models	L [1]	H [1]	A [1]	Pre-Training Tasks	Pre-Training Corpora
BERT [12]	12	768	12	Masked Language Model & NSP	BooksCorpus & Wikipedia
ERNIE-base [13]	12	768	12	Continual Multi-task Learning & NSP	Chinese Wikipedia, Baidu Baike, News, Tieba
ERNIE-large [13]	24	1024	16	Continual Multi-task Learning & NSP	Chinese Wikipedia, Baidu Baike, News, Tieba

[1] We denote the number of transformer layers as L, the hidden size as H, and the number of self-attention heads as A.

3. Methods

This section will specify how to perform the RE task based on the fusion of sequential and structural features using ERNIE.

3.1. Problem Definition

For the RE task, the input is the sequence containing named entity pairs. Thus, the RE task can be seen as a kind of text classification task, i.e., the category of the input sequence can be regarded as the relation of the entity pairs it contains. Given the dataset $D = \{(E_1, E_2), S, Y\}$, where E_1 and E_2 denote the set of entity pairs, S denotes the set of training texts, and Y denotes the set of relations between entity pairs. For any $e_i \in E_1$, $e_m \in E_2$, $s_k \in S$, and $y_l \in Y$, if there exists a relation y_l between e_i and e_m in the text s_k, it is denoted as $d_{i,m,k,l} \in D$. The mapping function $f(e_i, e_m, s_k) \rightarrow y_l$ is computed through the dataset D. For another dataset $D' = \{(E_1', E_2'), S', Y'\}$ with the same distribution as D, there exists $d'_{i,m,k,l} = \{(e_i', e_m'), s_k', y_l'\}$ and $d'_{i,m,k,l} \in D'$. The $\hat{y}_l' = f(e_i', e_m', s_k')$ should be as

close as possible to the true value y'_l. Therefore, the key to the RE task is to find a suitable model to determine the mapping function $f(e_i, e_m, s_k) \rightarrow y_l$ [36].

3.2. Model Architecture

In this subsection, we will first describe how to handle the sequential data using the pre-training model. Then, we will illustrate how to generate structural data based on sequential data and concatenate them as input to the model. Finally, how to perform the RE task based on the fusion of the features for these two types of data will be presented.

3.2.1. Sequential Data

As shown in Figure 3, the part of the input sequence marked by the light blue background is the sequential data. In this subsection, we assume that the only inputs to the model are sequential texts and try to improve the model from both input and output perspectives.

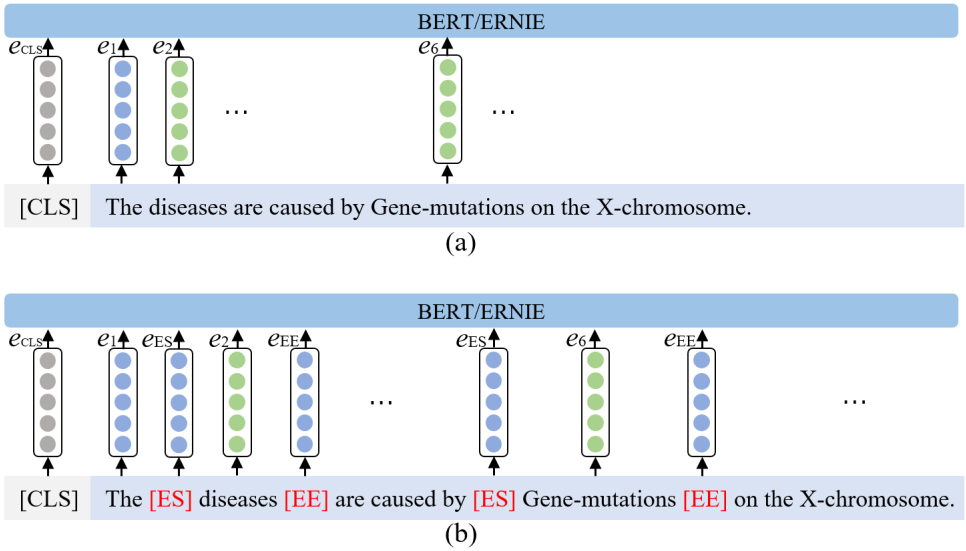

Figure 3. The difference between the common input sequence and the enhanced input sequence with entity markers. (**a**) The common input sequence. (**b**) The enhanced input sequence with entity markers.

Methods for Improving the Input Section

Different from the text classification task, the sequence of the RE task contains the entity pairs to be classified. Therefore, the model should not treat the entities as common tokens. An approach is needed that satisfies the following requirements: (1) "highlight" the entities in the input sequence; and (2) explicitly obtain the location information of the entities.

To this end, we explicitly add the entity position markers around the entities in the raw text. As shown in Figure 3a, the ordinary input sequence is "*The diseases are caused by Gene-mutations on the X-chromosome.*", which contains two named entities: "diseases" and "Gene-mutations". This sentence is extracted from the SemEval 2010 Task 8 dataset [37]. In contrast, as shown in Figure 3b, the entity pairs in the enhanced input sequence are highlighted with additional markers, i.e., `[ES]` and `[EE]`. The `[ES]` is the abbreviation of "Entity Start", indicating that the current token is the start of an entity. Similarly, the `[EE]` stands for "Entity End", indicating that the current token is the end of an entity. In addition, we also attempt to add entity embeddings to the input embeddings, as shown in Figure 4. In the experimental part of this paper, we will test the effectiveness of the above methods.

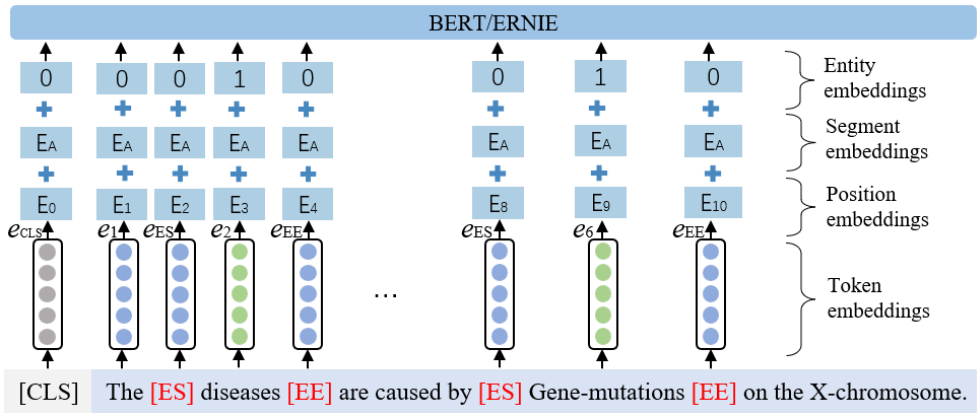

Figure 4. The entity-type embeddings are added to the input embeddings.

Methods for Improving the Output Section

The sequences with additional entity markers are fed into the pre-training model, and the representations corresponding to each token are generated after 12 layers of Transformer blocks [35]. It is worth noting that the `[CLS]` must be added at the beginning of a sequence before entering the model, as specified in the original paper of BERT and ERNIE, where the representation of the `[CLS]` is considered as the sentence-level feature [12,13]. In this case, as shown in Figure 5, for sequential data of length n, the embeddings can be denoted as $E = \{e_{CLS}, e_1, e_2, \ldots, e_n\}$, and the representations (or features) generated by the pre-training model can be denoted as $H = \{h_{CLS}, h_1, h_2, \ldots, h_n\}$, where the representations of `[CLS]` and `[ES]` are h_{CLS} and h_{ES}. In the following parts of this subsection, we will propose two methods to determine the relation for an entity pair. These two methods will be further compared in the section detailing the ablation experiment.

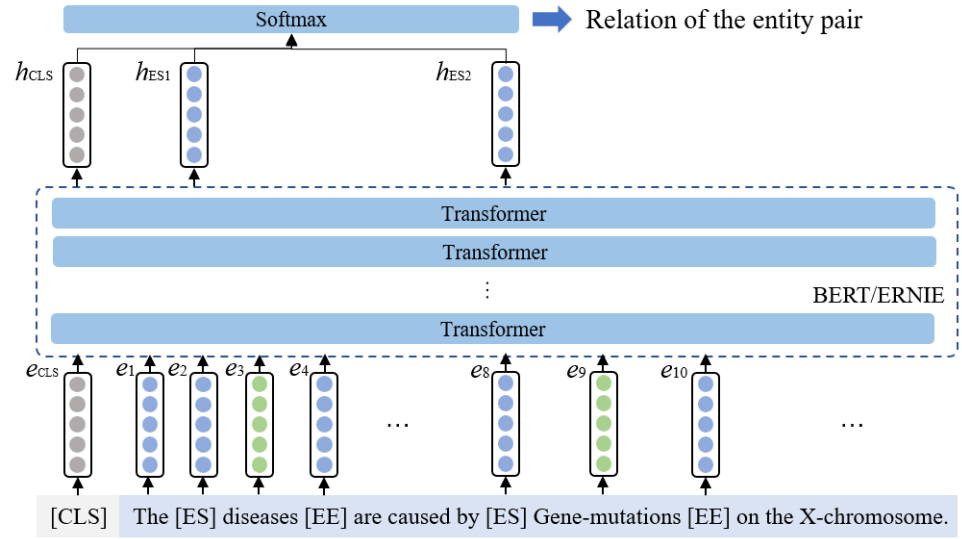

Figure 5. The relation label is generated based on the representations of sequential data.

Based on the Feature of `[CLS]`

As mentioned above, the representation of `[CLS]` can be regarded as the sentence-level feature. The paper proposing BERT also elaborates that this type of feature can be used for text classification. Since the RE task can essentially be considered as a kind of text classification task, we start the RE task with the sentence-level feature. At this point, the probability distribution P_C of the relation labels can be calculated by the following equation:

$$P_C = softmax(\mathbf{W}_C h_{CLS} + \boldsymbol{b}_C) \quad (1)$$

where $\mathbf{W}_C \in \mathbb{R}^{M*H}$ and $\mathbf{b}_C \in \mathbb{R}^M$ are the learnable weight matrix and bias, respectively. M is the number of relation labels, and H is the dimension of the hidden layer. Finally, the predicted label y_C can be obtained by the following equation:

$$y_C = argmax(P_C). \tag{2}$$

The loss function of this task is

$$loss = \sum_{m=1}^{M} p(y_C^m) \log[q(y_C^m)] \tag{3}$$

where $p(y_d^m)$ is the probability distribution of correct labels, and $q(y_d^m)$ is the probability distribution of predicted labels. The goal of this task is to minimize the loss function.

Based on the Features of `[CLS]` and `[ES]`

The idea behind this approach stems from combining the representation of `[CLS]` with those of other special tokens to generate features with even richer semantics. Specifically, we concatenate the representations of `[ES]`, which contain information about the location of the entity, with the representation of `[CLS]` to form a new feature vector. This feature vector is passed sequentially through the fully connected layer and the softmax layer to predict the relation label. As shown in Figure 5, h_{CLS} is the representation corresponding to the `[CLS]` token, while h_{ES1} and h_{ES2} are the representations of the two entity start position tokens [ES], respectively. H_S is constructed by concatenating these three features according to the following equation:

$$H_S = concat(h_{CLS} + h_{ES1} + h_{ES2}). \tag{4}$$

Once the new feature vector of sequential data H_S is obtained, the following calculations are performed in the same way as in the previous method. The probability distribution P_S for each label is calculated using the following equation:

$$P_S = softmax(\mathbf{W}H_S + \boldsymbol{b}). \tag{5}$$

Similarly, $\mathbf{W} \in \mathbb{R}^{M*H}$ and $\mathbf{b} \in \mathbb{R}^M$ are the learnable weight matrix and bias, respectively. M is the number of relations, and H is the dimension of the hidden layer. The final predicted label y_S is obtained through the following equation:

$$y_S = argmax(P_S). \tag{6}$$

We will also compare these two methods for the performance of relation extraction in the section on the ablation experiment.

The content above presents how we handle the sequential data to extract entity relations. Then we will describe how to construct structural data based on the sequential data and perform the RE task through the fusion of these two types of data.

3.2.2. Structural Data

In order to perform the RE task based on the fusion of sequential and structural data, a structural data-generation approach is required. This approach should not lead to additional training sets or annotations. Therefore, in this paper, the structural data are constructed through the dependency tree and the shortest dependency path based on the raw sequential data.

The dependency tree for the raw text is generated by a Python package called spaCy (https://github.com/explosion/spaCy, accessed on 18 January 2023). As mentioned earlier, the RE task is more concerned with the information regarding entity pairs. Therefore, we prune the dependency tree, leaving only the shortest dependency path. In the example given in Figure 6, the paths marked by the red arrows form the shortest dependency path.

In this way, the structural data "*diseases caused by Gene-mutations*" are constructed based on the raw sequential data.

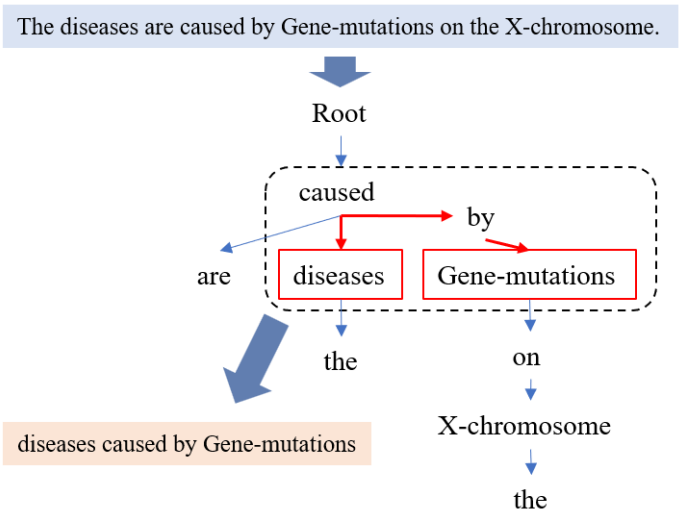

Figure 6. The structural data are constructed using the dependency tree and the shortest dependency path based on the raw sequential data.

3.2.3. Fusion of Sequential and Structural Features

In order to take advantage of the fusion of sequential and structural data, the first issue that needs to be addressed is how to feed the acquired structural data into the model. Figure 7 shows the final architecture of the model proposed in this paper, which is an improved version of the model proposed in Section 3.2.1. For the input to this model, sequential data and structural data are concatenated with the `[SEP]` token to form the new input sentence.

This method of concatenating the two types of data with `[SEP]` is inspired by BERT for the sentence pair classification task [12]. The input sequences to this task are all sentence pairs concatenated with `[SEP]`, which is consistent with the task in this study since the input consisting of sequential and structural data can also be regarded as the sentence pair. The next question that needs to be addressed is which representations are chosen to compute the distribution probabilities of the relation labels.

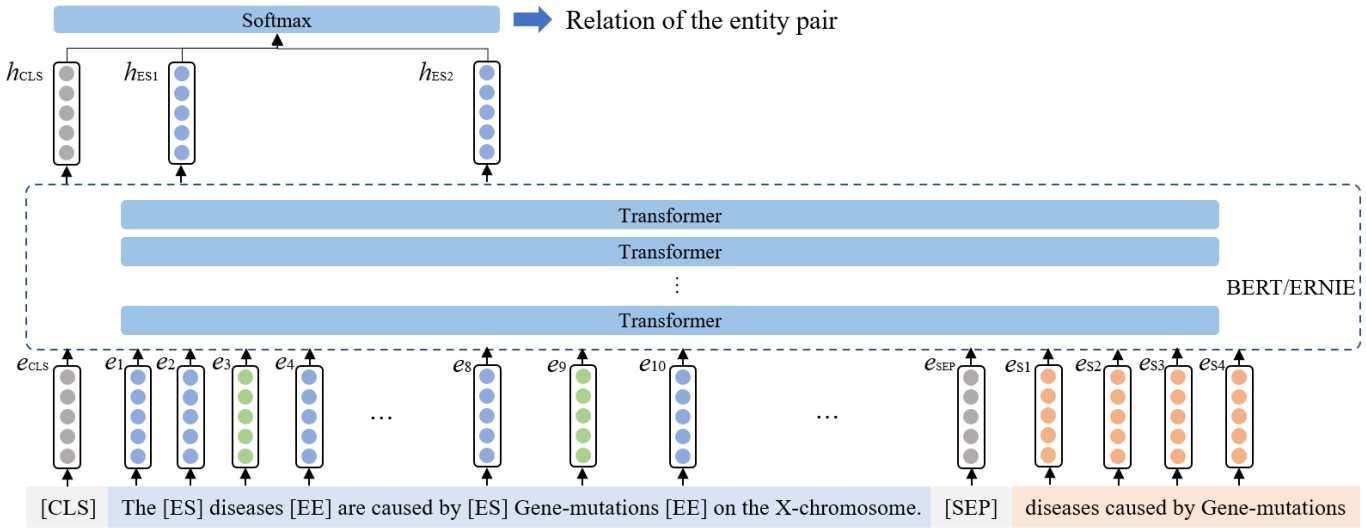

Figure 7. The final architecture of the proposed model. For the input of this model, sequential data and structural data are concatenated with the `[SEP]` token to form the new input sentence.

Furthermore, inspired by BERT, which uses the representation of [CLS] to handle sentence pair classification tasks, such as MNLI (multi-genre natural language inference), QQP (Quora question pairs), and QNLI (question–answer natural language inference) [38], one can speculate that the representation of [CLS] contains the semantic features of the sentence pair, or one can say that in the model proposed in this paper, the representation of [CLS] can be regarded as a kind of hybrid feature. The subsequent computational process is consistent with Section 3.2.1, and the representation of [CLS] containing the semantic information is concatenated with the representations of [ES] according to the following equation:

$$H = concat(h_{CLS} + h_{ES1} + h_{ES2}). \tag{7}$$

The probability distribution P for each label is obtained by the following equation:

$$P = softmax(\mathbf{W}H + \boldsymbol{b}). \tag{8}$$

Similarly, $\mathbf{W} \in \mathbb{R}^{M*H}$ and $\mathbf{b} \in \mathbb{R}^{M}$ are the learnable weight matrix and bias, respectively. M is the number of relations, and H is the dimension of hidden layers. The final predicted label y can be generated by the following equation:

$$y = argmax(P). \tag{9}$$

The above is the architecture of the model proposed in this paper, which leverages a fusion of sequential and structural features for the RE task. In the following subsection, we will validate the performance of the model using the SemEval 2010 Task 8 dataset.

4. Experiments and Results

The software environment for this experiment is Paddlepaddle (https://github.com/paddlepaddle/paddle, accessed on 18 January 2023), which is an end-to-end open-source deep learning platform developed by Baidu, and the hardware environment is an 8-core CPU with an Nvidia V100 GPU.

4.1. Dataset

The dataset adopted for this experiment is "SemEval 2010 Task 8". There are nine types of relations contained in this dataset: "Cause Effect", "Component Whole", "Content Container ", "Entity Destination", "Entity Origin", "Instrument Agency ", "Member Collection", "Message Topic", and "Product Producer". When the relation for an entity pair does not belong to the above nine labels, it is classified as "Other". The distribution of relation labels in the dataset is shown in Table 3.

Table 3. The distribution of labels in the SemEval 2010 Task 8 dataset.

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

Each instance in the dataset contains two entities, called "Entity1" and "Entity2", and the relations for these entity pairs are directional, i.e., the graph composed of these entity

pairs with relations is a directed graph from the perspective of the knowledge graph. For example, the two graph units shown in Figure 8 are both composed of entity nodes named Entity1 and Entity2 but their directions are different. Hence, they are two different kinds of relations. For this reason, every relation label in Table 3 is considered with direction in the experiment, except the label "Other".

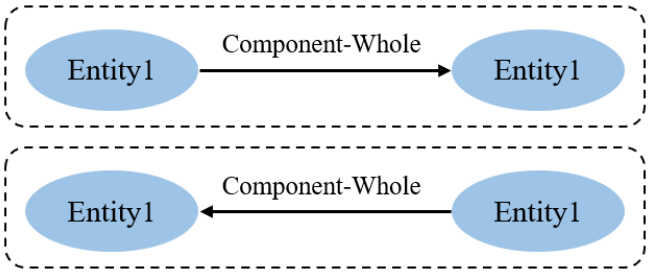

Figure 8. The relation of entity pairs is directional.

4.2. Hyperparameters

The hyperparameters involved in this experiment are listed in Table 4, and the optimizer we used is Adam.

Table 4. Hyperparameters.

Hyperparameters	Values
Epoch	10
Learning rate	5×10^{-5}
Input max length	128
Batch size	64

4.3. Evaluation Metrics

We introduce the *F1* value to evaluate the performance of the models listed in Table 5. This metric is calculated according to the following formulation, where the precision value refers to the ratio of correct entities to predicted entities, and the recall value is the proportion of the entities in the test set that are correctly predicted.

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{10}$$

4.4. Results of Different Models

This experiment first compares the proposed model with machine learning and deep learning methods, such as SVM, MVRNN, and CR-CNN. The results of these models are listed in the first block of Table 5. As can be seen from the results, the machine learning method (i.e., the SVM-based method [18]) obtains the lowest *F1* value of 0.822. In contrast, the deep learning methods, both RNN-based and CNN-based, are able to improve the *F1* value. For example, the MVRNN [39] obtains an *F1* value of 0.824, while the CR-CNN [40] obtains an *F1* value of 0.841. The FAT-RE [41] based on the dependency tree obtains an *F1* value of 0.842. In addition, the hybrid method BiLSTM+CNN [42] improves the *F1* value to 0.854. Introducing the attention mechanism also boosts the performance of deep learning models. For example, Att-RCNN [43] obtains an *F1* value of 0.866, which is second only to the methods based on the pre-training models.

Table 5. The results of different models.

Models	*F1* Values
SVM [18]	0.822
MVRNN [39]	0.824
CNN+Softmax [10]	0.827
FCM [44]	0.830
CR-CNN [40]	0.841
FAT-RE [41]	0.842
CNN [45]	0.848
TACNN [46]	0.853
BiLSTM+CNN [42]	0.854
Att-RCNN [43]	0.866
BERT-tokens (REflex) [30]	0.867
BERT-entity (OpenNRE) [16]	0.883
E-BEM [47]	0.885
E-BEM-O [47]	0.886
BERT-DP [17]	0.891
NLIRE [48]	0.894
Our model (BERT)	0.893
Our model (ERNIE-large)	0.902

As previously mentioned, pre-training models perform better in various downstream NLP tasks leveraging the prior syntactic and semantic knowledge acquired in the large-scale unlabeled corpus. The results of these approaches are listed in the second block of Table 5, and these methods focus more on sequential data. For instance, the BERT-tokens, as a module in REflex [30], concatenate the word-piece-level embeddings provided by BERT with the word and position embeddings of CRCNN before entering the convolution phase. The *F1* value obtained by this module is slightly higher than that of the deep learning method based on the attention mechanism, reaching 0.867. The OpenNRE is an open-source toolkit for relation extraction, and the BERT entity is one of the components used for sentence-level relation extraction [16]. The implementation concept of BERT-entity is analogous to that of this paper for sequential data, and the component receives an *F1* value of 0.883. The E-BEM and E-BEM-O are both extended versions of BERT-entity [47], and increase the *F1* value to 0.885 and 0.886, respectively. Some studies focus on the use of structured data to handle the relation extraction task. For example, BERT-DP [17] obtains an *F1* value of 0.891 by feeding the positional encoding with a dependency tree into BERT. In addition, NLIRE is an RE model based on natural language inference and obtains an *F1* value of 0.894. However, the model we proposed (based on ERNIE, which performs the relation extraction task based on the fusion of sequential and structural features) obtains the highest *F1* value, as shown in Table 5. For the purpose of comparison with previous work, we also present the results obtained by our model based on BERT.

4.5. Results Using Different Pre-Training Models

In this experiment, we test the effect of selecting different pre-training models on the performance based on the structure described in Section 3.2.3. As shown in Table 6, the accuracy, recall, and *F1* value can be further improved using ERNIE series models compared with BERT. Specifically, ERNIE-base improves the *F1* value by 0.3 percentage points compared with BERT, while ERNIE-large achieves the highest *F1* value of 0.902.

In addition, we also compared the *F1* values of the different pre-training models at each training epoch. As can be seen in Figure 9, the sorting based on *F1* values in each epoch is consistent with the final results, except for the third epoch. This phenomenon indicates that ERNIE-large converges better than other models at the beginning of the training process.

Table 6. The results of different pre-training models.

Models	Accuracy	Recall	*F1* Values
BERT-[ES] & SDP-[CLS] & [ES]	0.880	0.906	0.893
ERNIE-base-[ES] & SDP-[CLS] & [ES]	0.883	0.911	0.897
ERNIE-large-[ES] & SDP-[CLS] & [ES]	0.888	0.917	0.902

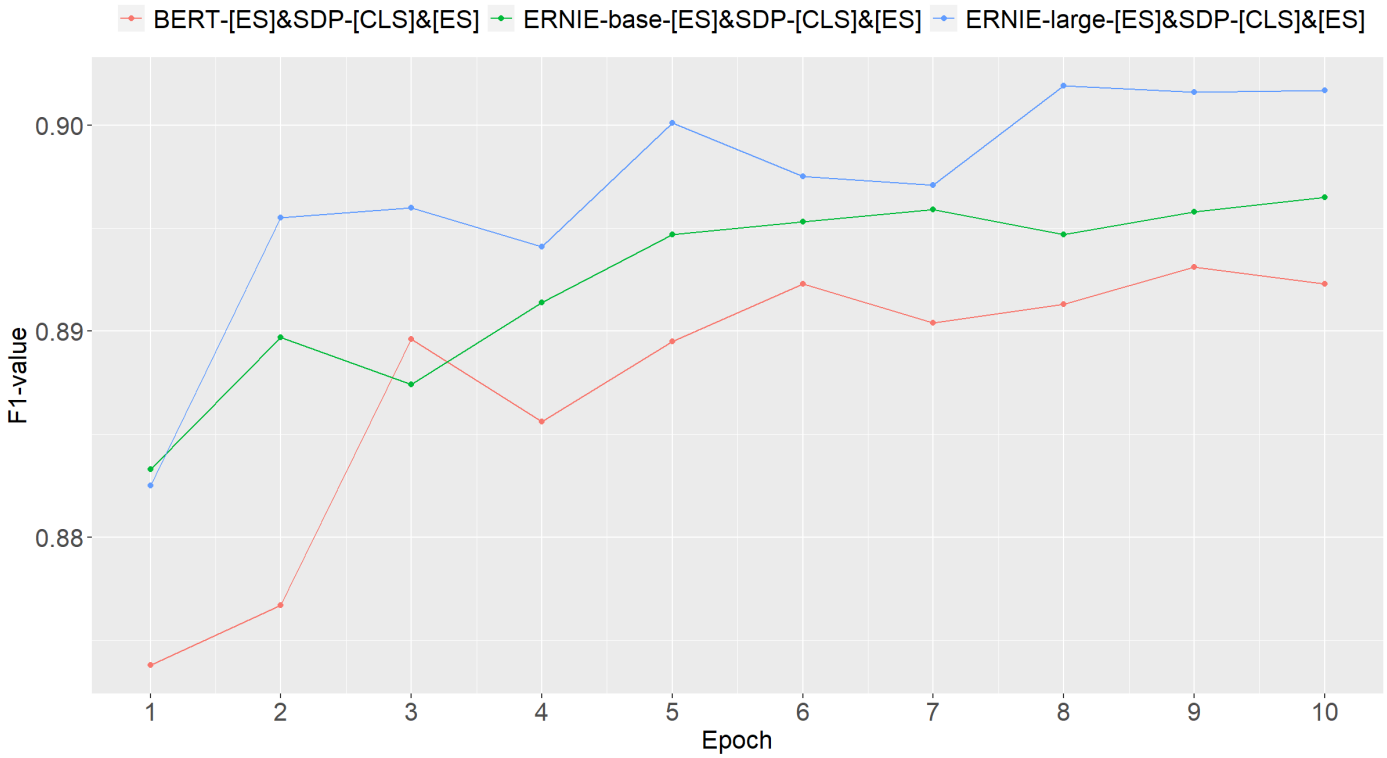

Figure 9. The *F1* values using different pre-training models at each training epoch.

4.6. Ablation Experiment

We also performed an ablation experiment. The purpose of this experiment is to verify a component by removing it from the proposed model. As shown in Table 7, using the raw text and classifying the relations based on the features of `[CLS]` can obtain a higher *F1* value compared with the machine learning methods listed in the first block of Table 5. This approach was also proposed by Devlin et al. [12], who applied BERT to the single-sentence classification task. In view of this, this approach is also selected as the benchmark in this paper. Furthermore, both adding `[ES]` and `[EE]` tokens to the raw text and concatenating the features of `[CLS]` and `[ES]` to compute the probability distribution can improve the performance. However, it is worth mentioning that the entity embeddings do not contribute to the improvement of the *F1* value. The *F1* value can be further enhanced based on the fusion of sequential and structural features. Finally, as the results presented in Table 5 show, the highest recall and *F1* value are obtained when ERNIE-large is chosen for the pre-training model in our proposed approach.

In addition, we also recorded *F1* values for each training epoch in the ablation experiment. As shown in Figure 10, the general trend of the curves is identical to the results shown in Table 7, i.e., the *F1* value can be gradually increased by joining the corresponding components. From the lower two figures of Figure 10, it can be seen that the RE models based on the fusion of sequential and structural features achieve higher *F1* values in each epoch, and they also converge faster in the early epochs of the training process. This experiment also demonstrates that the method proposed in this paper is efficient and effective.

Table 7. The results of the ablation experiment.

Models	Accuracy	Recall	*F1* Values
BERT-raw-[CLS] (benchmark)	0.881	0.890	0.885
BERT-[ES]-[CLS]	0.879	0.902	0.891
BERT-[ES][Entity embeddings]-[CLS]	0.875	0.906	0.891
BERT-[ES]-[CLS] & [ES]	0.879	0.907	0.892
BERT-[ES] & SDP-[CLS] & [ES]	0.880	0.906	0.893
ERNIE-base-[ES]-[CLS] & [ES]	0.878	0.910	0.894
ERNIE-base-[ES] & SDP-[CLS] & [ES]	0.883	0.911	0.897
ERNIE-large-[ES]-[CLS] & [ES]	0.890	0.909	0.899
ERNIE-large-[ES] & SDP-[CLS] & [ES]	0.888	0.917	0.902

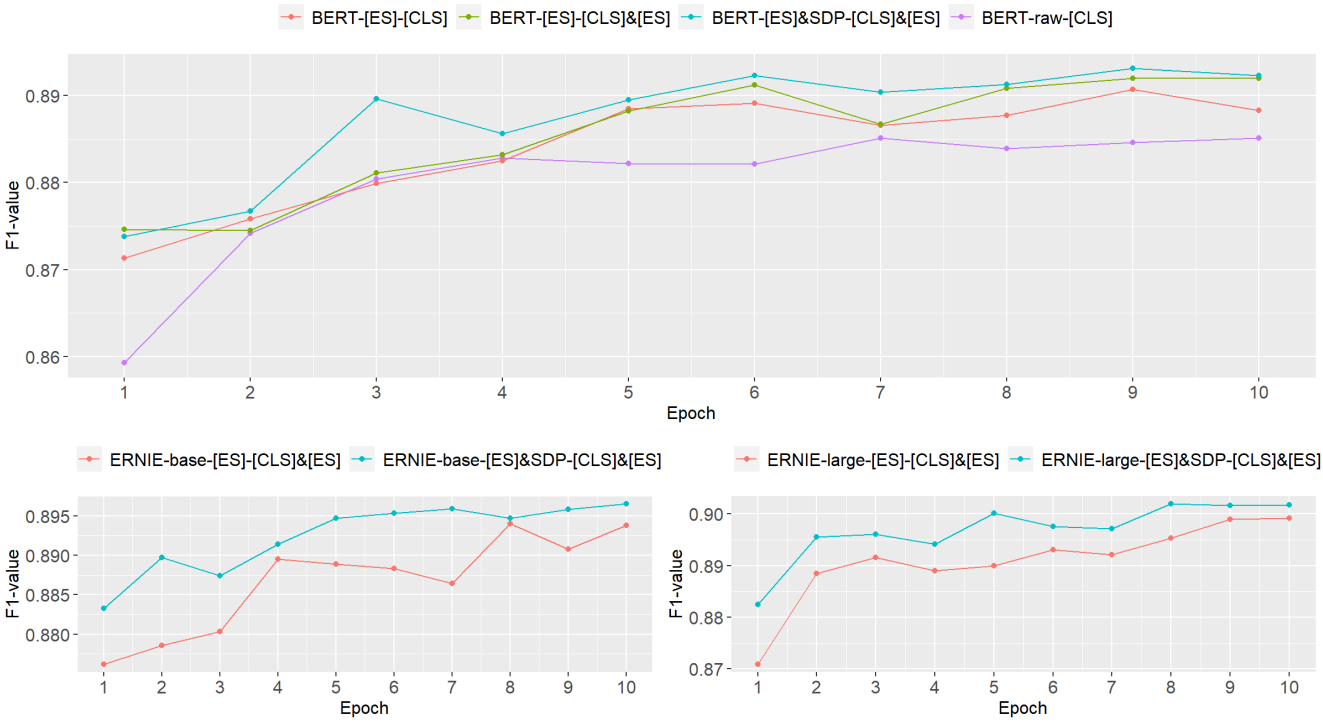

Figure 10. The *F1* values of the ablation experiment at each training epoch.

5. Discussion

Firstly, the results presented in Table 5 demonstrate that the proposed method outperforms previous approaches based on machine learning, deep learning, and pre-training models. As previously mentioned, these supervised learning methods based on machine learning and deep learning can only capture the semantic knowledge in the labeled training samples, which are difficult to access. The pre-training models, on the other hand, can acquire extensive semantic and syntactic knowledge from the massive unlabelled corpus through the pre-training process and leverage this knowledge to improve the performance of downstream NLP tasks. Thus, the methods based on pre-training models in the second block of Table 5 all outperform those based on machine learning and deep learning listed in the first block. However, these RE methods designed upon pre-training models only consider the sequential or structural features individually, ignoring the fact that fusing these two features can enhance the performance. Instead, the model we propose generates the fusion of these two features by utilizing the representation of `[CLS]`, and applying them to the RE task. According to Devlin et al. [12], when conducting the multi-sentence classification task using BERT, it is necessary to concatenate two sentences with `[SEP]` and obtain the semantic features, leveraging the representation of `[CLS]`. Therefore, in such

a task scenario, the representation of [CLS] would naturally contain the semantics of the input sentence pairs. Returning to the goal of this study, as described in Section 3.2.3, the sequential and structural data are concatenated with [SEP] as the input sentence. At this point, the representation of [CLS] can be regarded as a fusion of sequential and structural features. It is worth noting that the pre-training model chosen for this experiment is BERT, in order to allow comparison with previous work. The experimental results also show that leveraging this enhanced representation can improve the *F1* value to 0.893.

Secondly, Table 5 illustrates the *F1* value when using different pre-training models based on the architecture proposed in Section 3.2.3. The results indicate that the accuracy, recall, and *F1* value can be further improved by using the ERNIE series models, while the highest accuracy, recall, and *F1* value can be obtained by using ERNIE-large. This phenomenon demonstrates that: (1) For the same structure of the pre-training models, the pre-training process is critical to obtain appropriate prior semantic knowledge. The BERT and ERNIE-base have the same structure, both consisting of 12-layer Transformer blocks and 12 self-attention heads. However, the ERNIE-base can acquire richer semantic knowledge than BERT through multi-task learning consisting of word-aware, structure-aware, and semantic-aware tasks. (2) For pre-training models with the same pre-training task, such as ERNIE-base and ERNIE-large, the more complex the model structure, the richer the prior semantic knowledge it can capture. The base model contains 12-layer Transformer blocks, 12 self-attention heads, and a 768-dimension hidden size while the large model contains 24-layer Transformer blocks, 16 self-attention heads, and a 1024-dimension hidden size. Finally, the ERNIE-large model has the highest *F1* value of 0.902. Figure 9 also proves the above conclusion in another regard. Since the fourth epoch, the *F1* values of the ERNIE series model are consistently higher than those of the BERT, and the ERNIE-large achieves the highest *F1* values at each epoch.

Thirdly, the results of the ablation experiment demonstrate that adding the [ES] to the input sequence can explicitly introduce the position information of the entity pairs. Meanwhile, for the sequential data, the representation of [ES] carries the relevant information of entity pairs. As described earlier, the RE task is more concerned with the information of entity pairs than the text classification task. Therefore, leveraging the representation of [ES] can enhance the effectiveness of classification. However, it is worth mentioning that the entity embedding does not contribute to the improvement of the *F1* value, although it slightly increases the recall. We assume that a possible reason could be that the introduction of additional embeddings would increase the complexity of our model. Finally, by using the fusion method proposed in this study, both sequential and structural features can be used simultaneously to obtain the highest recall and *F1* value. Table 7 and Figure 10 illustrate that adding the relevant modules can indeed gradually increase the *F1* values.

In summary, the advantages of the proposed method are as follows:

(1) Inspired by a pre-training task called NSP, we design a concise relation extraction method modeled upon it. The method conducts the RE task through a fusion of sequential and structural features. The pruned SDP obviously contains structural information, which improves the *F1* value compared with the approaches leveraging only the sequential feature, as demonstrated by the results of Table 7 and the ablation experiment.

(2) For sequential data, we enhance the input and output sides of the model separately. Table 7 and the ablation experiment also show that the sequential feature can contain information about entity locations by adding the location makers and using the representations corresponding to the ES. Given that the RE task focuses more on the entity pairs in the input sequence, this operation can boost the *F1* value.

(3) The proposed model is constructed based on a pre-training model, which is pre-trained on a large-scale corpus, and can generate the representations with prior semantic knowledge. As shown in Table 5, all of the methods based on the pre-training model outperform the methods based on machine learning. In addition, for the selection of the pre-training model, we choose ERNIE-large because it can acquire richer semantic knowledge than BERT through multi-task learning and more parameters.

However, our approach still has some drawbacks, which are as follows:

(1) We chose the ERNIE-large model with more parameters; the length of the input sentences increases when both the sequential and structural data are included. Therefore, in terms of efficiency, our method may be slightly less efficient than previous methods.

(2) The method for generating the SDP requires a third-party toolkit and its errors may be introduced into the relational classification task.

We hope to address these drawbacks in future work to enhance the effectiveness and accuracy of our model.

6. Conclusions

Relation extraction, as an important NLP task, aims to identify the relation between two entities within unstructured data. The pre-training models capture the prior semantic knowledge from unlabeled corpora and leverage it to enhance the downstream NLP tasks, including the RE task. However, previous research has mainly focused on sequential or structural data separately, ignoring the fact that fusing sequential and structural features may enhance the performance of the RE task. In this study, we explored how to use the fused features for the RE task based on pre-training models.

Firstly, for the sequential data, we inserted the entity location tokens `[ES]` and `[EE]` into the raw text and try to add the entity embeddings to the input embeddings; for the output side, we experimented with the performance when applying the representations of `[CLS]` and `[ES]` on the RE task. Secondly, we propose a concise RE approach based on the fusion of sequential and structural information using a pre-training model. Inspired by the pre-training task NSP, we use the `[SEP]` token to concatenate the shortest dependent paths of entity pairs with the original sequence and then obtain the representation of the `[CLS]` token, which can be regarded as the fusion of sequential and structural features. The experiments were conducted on the SemEval 2010 Task 8 dataset, and the results show that the proposed method can improve the *F1* value to 0.902.

Author Contributions: Conceptualization, Y.W. (Yu Wang) and Y.W. (Yuan Wang); methodology, Y.W. (Yu Wang); software, Y.W. (Yuan Wang); validation, Z.P.; formal analysis, Z.P.; investigation, Y.W. (Yu Wang); resources, F.Z.; data curation, F.Z.; writing—original draft preparation, Y.W. (Yu Wang); writing—review and editing, F.Y.; supervision, F.Y.; project administration, Y.W. (Yuan Wang); funding acquisition, F.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Natural Science Foundation of Anhui Province of China (no.2108085 MH303, no. 2108085QF274) and the Initiation Fund of Anhui Medical University (no. 1401039201).

Data Availability Statement: The original dataset used in this study is publicly available at https://paperswithcode.com/sota/relation-extraction-on-semeval-2010-task-8 (accessed on 18 January 2023).

Acknowledgments: This research was supported by the Medical Big Data Supercomputing Center System of Anhui Medical University.

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

Abbreviations

The following abbreviations are used in this manuscript:

BERT	bidirectional encoder representation from transformers
BiLSTM	bidirectional long short-term memory
CNN	convolutional neural network
EE	entity end
ES	entity start
ERNIE	enhanced representation through knowledge integration
GPU	graphics processing unit
KG	knowledge graph

KNN	K-nearest neighbors
MNLI	multi-genre natural language inference
MSRA	Microsoft Research Asia
NER	named entity recognition
NLP	natural language processing
NSP	next sentence prediction
QNLI	question–answer natural language inference
QQP	Quora question pairs
RNN	recurrent neural network
RE	relation extraction
SDP	shortest dependency path
SVM	support vector machine

References

1. Sboev, A.; Rybka, R.; Selivanov, A.; Moloshnikov, I.; Gryaznov, A.; Naumov, A.; Sboeva, S.; Rylkov, G.; Zakirova, S. Accuracy Analysis of the End-to-End Extraction of Related Named Entities from Russian Drug Review Texts by Modern Approaches Validated on English Biomedical Corpora. *Mathematics* **2023**, *11*, 354. [CrossRef]
2. Lezama-Sánchez, A.L.; Tovar Vidal, M.; Reyes-Ortiz, J.A. An Approach Based on Semantic Relationship Embeddings for Text Classification. *Mathematics* **2022**, *10*, 4161. [CrossRef]
3. Wang, H.; Zhang, F.; Xie, X.; Guo, M. DKN: Deep knowledge-aware network for news recommendation. In Proceedings of the 2018 World Wide Web Conference, Lyon, France, 23–27 April 2018; pp. 1835–1844.
4. Sun, Q.; Xu, T.; Zhang, K.; Huang, K.; Lv, L.; Li, X.; Zhang, T.; Dore-Natteh, D. Dual-Channel and Hierarchical Graph Convolutional Networks for document-level relation extraction. *Expert Syst. Appl.* **2022**, *205*, 117678. [CrossRef]
5. Le, H.Q.; Can, D.C.; Collier, N. Exploiting document graphs for inter sentence relation extraction. *J. Biomed. Semant.* **2022**, *13*, 1–15. [CrossRef] [PubMed]
6. Zhou, D.; Zhong, D.; He, Y. Biomedical relation extraction: From binary to complex. *Comput. Math. Methods Med.* **2014**, *2014*, 298473. [CrossRef]
7. Lai, P.T.; Lu, Z. BERT-GT: Cross-sentence n-ary relation extraction with BERT and Graph Transformer. *Bioinformatics* **2020**, *36*, 5678–5685. [CrossRef]
8. Bui, Q.C.; Sloot, P.M.; Van Mulligen, E.M.; Kors, J.A. A novel feature-based approach to extract drug–drug interactions from biomedical text. *Bioinformatics* **2014**, *30*, 3365–3371. [CrossRef] [PubMed]
9. 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 and Computational Natural Language Learning, Jeju Island, Republic of Korea, 12–14 July 2012; pp. 455–465.
10. 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.
11. 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.
12. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, 2–7 June 2019; Volume 1, pp. 4171–4186.
13. Sun, Y.; Wang, S.; Li, Y.; Feng, S.; Tian, H.; Wu, H.; Wang, H. Ernie 2.0: A continual pre-training framework for language understanding. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; pp. 8968–8975.
14. Wang, Y.; Sun, Y.; Ma, Z.; Gao, L.; Xu, Y.; Wu, Y. A method of relation extraction using pre-training models. In Proceedings of the 2020 13th International Symposium on Computational Intelligence and Design (ISCID), Hangzhou, China, 12–13 December 2020; pp. 176–179.
15. Wei, Q.; Ji, Z.; Si, Y.; Du, J.; Wang, J.; Tiryaki, F.; Wu, S.; Tao, C.; Roberts, K.; Xu, H. Relation extraction from clinical narratives using pre-trained language models. In Proceedings of the AMIA annual symposium proceedings. American Medical Informatics Association, Washington, DC, USA, 16–20 November 2019; Volume 2019, p. 1236.
16. Han, X.; Gao, T.; Yao, Y.; Ye, D.; Liu, Z.; Sun, M. OpenNRE: An Open and Extensible Toolkit for Neural Relation Extraction. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations, Hong Kong, China, 3–9 November 2019; pp. 169–174.
17. Cho, C.; Choi, Y.S. Dependency tree positional encoding method for relation extraction. In Proceedings of the 36th Annual ACM Symposium on Applied Computing, Virtual, 22–26 March 2021; pp. 1012–1020.

18. Rink, B.; Harabagiu, S. Utd: Classifying semantic relations by combining 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.
19. Zhao, S.; Grishman, R. Extracting relations with integrated information using kernel methods. In Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05), Ann Arbor, MI, USA, 25–30 June 2005; pp. 419–426.
20. Kim, S.; Liu, H.; Yeganova, L.; Wilbur, W.J. Extracting drug–drug interactions from literature using a rich feature-based linear kernel approach. *J. Biomed. Inform.* **2015**, *55*, 23–30. [CrossRef]
21. Choi, S.P. Extraction of protein–protein interactions (PPIs) from the literature by deep convolutional neural networks with various feature embeddings. *J. Inf. Sci.* **2018**, *44*, 60–73. [CrossRef]
22. Liu, S.; Tang, B.; Chen, Q.; Wang, X. Drug-drug interaction extraction via convolutional neural networks. *Comput. Math. Methods Med.* **2016**, *2016*, 6918381. [CrossRef]
23. Peng, Y.; Lu, Z. Deep learning for extracting protein-protein interactions from biomedical literature. In Proceedings of the BioNLP 2017, Vancouver, BC, Canada, 4 August 2017; pp. 29–38.
24. Zhou, H.; Ning, S.; Yang, Y.; Liu, Z.; Lang, C.; Lin, Y. Chemical-induced disease relation extraction with dependency information and prior knowledge. *J. Biomed. Inform.* **2018**, *84*, 171–178. [CrossRef] [PubMed]
25. Xu, Y.; Jia, R.; Mou, L.; Li, G.; Chen, Y.; Lu, Y.; Jin, Z. Improved relation classification by deep recurrent neural networks with data augmentation. In Proceedings of the COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, Osaka, Japan, 11–16 December 2016; pp. 1461–1470.
26. Li, F.; Zhang, M.; Fu, G.; Ji, D. A neural joint model for entity and relation extraction from biomedical text. *BMC Bioinform.* **2017**, *18*, 198. [CrossRef] [PubMed]
27. Wang, W.; Yang, X.; Yang, C.; Guo, X.; Zhang, X.; Wu, C. Dependency-based long short term memory network for drug-drug interaction extraction. *BMC Bioinform.* **2017**, *18*, 99–109. [CrossRef] [PubMed]
28. Xu, B.; Shi, X.; Zhao, Z.; Zheng, W. Leveraging biomedical resources in bi-lstm for drug-drug interaction extraction. *IEEE Access* **2018**, *6*, 33432–33439. [CrossRef]
29. Corbett, P.; Boyle, J. Improving the learning of chemical-protein interactions from literature using transfer learning and specialized word embeddings. *Database* **2018**, *2018*, bay066. [CrossRef]
30. Chauhan, G.; McDermott, M.B.; Szolovits, P. REflex: Flexible Framework for Relation Extraction in Multiple Domains. In Proceedings of the 18th BioNLP Workshop and Shared Task, Florence, Italy, 1 August 2019; pp. 30–47.
31. Pouyanfar, S.; Sadiq, S.; Yan, Y.; Tian, H.; Tao, Y.; Reyes, M.P.; Shyu, M.L.; Chen, S.C.; Iyengar, S.S. A survey on deep learning: Algorithms, techniques, and applications. *ACM Comput. Surv. (CSUR)* **2018**, *51*, 1–36. [CrossRef]
32. Zhang, Y.; Lin, H.; Yang, Z.; Wang, J.; Sun, Y.; Xu, B.; Zhao, Z. Neural network-based approaches for biomedical relation classification: A review. *J. Biomed. Inform.* **2019**, *99*, 103294. [CrossRef]
33. Cui, M.; Li, L.; Wang, Z.; You, M. A survey on relation extraction. In Proceedings of the China Conference on Knowledge Graph and Semantic Computing, Chendu China, 26 August 2017; pp. 50–58.
34. Liu, K. A survey on neural relation extraction. *Sci. China Technol. Sci.* **2020**, *63*, 1971–1989. [CrossRef]
35. 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.
36. Zhang, D.; Peng, D. ENT-BERT: Entity Relation Classification Model Combining BERT and Entity Information. *J. Chin. Comput. Syst.* **2020**, *41*, 2557–2562.
37. Hendrickx, I.; Kim, S.N.; Kozareva, Z.; Nakov, P.; Ó Séaghdha, D.; Padó, S.; Pennacchiotti, M.; Romano, L.; Szpakowicz, S. SemEval-2010 Task 8: Multi-Way Classification of Semantic Relations between Pairs of Nominals. In Proceedings of the 5th International Workshop on Semantic Evaluation, Los Angeles, CA, USA, 15–16 July 2010; pp. 33–38.
38. Wang, A.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; Bowman, S. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, Brussels, Belgium, 1 November 2018; pp. 353–355. [CrossRef]
39. Socher, R.; Huval, B.; Manning, C.D.; Ng, A.Y. Semantic compositionality through recursive matrix-vector spaces. In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Jeju Island, Republic of Korea, 12–14 July 2012; pp. 1201–1211.
40. Santos, C.; Bing, X.; Zhou, B. Classifying Relations by Ranking with Convolutional Neural Networks. *Comput. Sci.* **2015**, *86*, 132–137.
41. Ding, L.; Lei, Z.; Xun, G.; Yang, Y. FAT-RE: A faster dependency-free model for relation extraction. *J. Web Semant.* **2020**, *65*, 100598. [CrossRef]
42. Wang, Y.; Han, Z.; You, K.; Lin, Z. A Two-channel model for relation extraction using multiple trained word embeddings. *Knowl.-Based Syst.* **2022**, *255*, 109701. [CrossRef]
43. Guo, X.; Zhang, H.; Yang, H.; Xu, L.; Ye, Z. A Single Attention-Based Combination of CNN and RNN for Relation Classification. *IEEE Access* **2019**, *7*, 12467–12475. [CrossRef]
44. Yu, M.; Gormley, M.; Dredze, M. Factor-based compositional embedding models. In Proceedings of the NIPS Workshop on Learning Semantics, Montreal, QC, Canada, 8–11 December 2014; pp. 95–101.

45. Qin, P.; Xu, W.; Guo, J. An empirical convolutional neural network approach for semantic relation classification. *Neurocomputing* **2016**, *190*, 1–9. [CrossRef]
46. Geng, Z.; Li, J.; Han, Y.; Zhang, Y. Novel target attention convolutional neural network for relation classification. *Inf. Sci.* **2022**, *597*, 24–37. [CrossRef]
47. Nascimento, I.; Lima, R.; Chifu, A.; Espinasse, B.; Fournier, S. DeepREF: A Framework for Optimized Deep Learning-based Relation Classification. In Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022), Palais du Pharo, France, 20–25 June 2022; pp. 4513–4522.
48. Hu, W.; Liu, L.; Sun, Y.; Wu, Y.; Liu, Z.; Zhang, R.; Peng, T. NLIRE: A Natural Language Inference method for Relation Extraction. *J. Web Semant.* **2022**, *72*, 100686. [CrossRef]