symmetry

MDPI

Article

Distantly Supervised Relation Extraction via Contextual Information Interaction and Relation Embeddings

Huixin Yin, Shengquan Liu * and Zhaorui Jian

College of Information Science and Engineering, Xinjiang University, Urumqi 830046, China; 107552103573@stu.xju.edu.cn (H.Y.); 107552101302@stu.xju.edu.cn (Z.J.)
* Correspondence: liu@xju.edu.cn

Abstract: Distantly supervised relation extraction (DSRE) utilizes an external knowledge base to automatically label a corpus, which inevitably leads to the problem of mislabeling. Existing approaches utilize BERT to provide instances and relation embeddings to capture a wide set of relations and address the noise problem. However, the method suffers from a single method of textual information processing, underutilizing the feature information of entity pairs in the relation embeddings part and being interfered with by noisy labels when classifying multiple labels. For this reason, we propose the contextual information interaction and relation embeddings (CIRE) method. First, we utilize BERT and Bi-LSTM to construct a neural network model to enhance contextual information interaction by filtering and supplementing sequence information through the error repair capability of the Bi-LSTM gating mechanism. At the same time, we combine the vector difference between entity pairs and entity pairs in the relation embeddings layer to improve the relation embeddings accuracy. Finally, we choose sparse softmax as the classifier, which improves the ability to control the noise categories by controlling the number of output categories. The experimental results show that our method significantly outperforms the baseline method and improves the AUC metric by 2.6% on the NYT2010 dataset.

Keywords: distantly supervised relation extraction; neural network model; relation embeddings; sparse softmax

check for updates

Citation: Yin, H.; Liu, S.; Jian, Z. Distantly Supervised Relation Extraction via Contextual Information Interaction and Relation Embeddings. *Symmetry* **2023**, *15*, 1788. https://doi.org/10.3390/sym15091788

Academic Editor: Lorentz Jäntschi

Received: 29 August 2023
Revised: 15 September 2023
Accepted: 17 September 2023
Published: 18 September 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

DSRE is a method first proposed by Mintz et al. [1]. The approach is based on the assumption that if two entities exhibit a relationship in the knowledge base, then all sentences referring to these two entities will be labeled as such. This assumption is too strong and inevitably leads to some noisy labeling. Take Figure 1 as an example. The distantly supervised (DS) approach labels all sentences containing "Steve Jobs" and "Apple" as "founder" relationship types, but the third sentence in the figure does not express such a relationship.

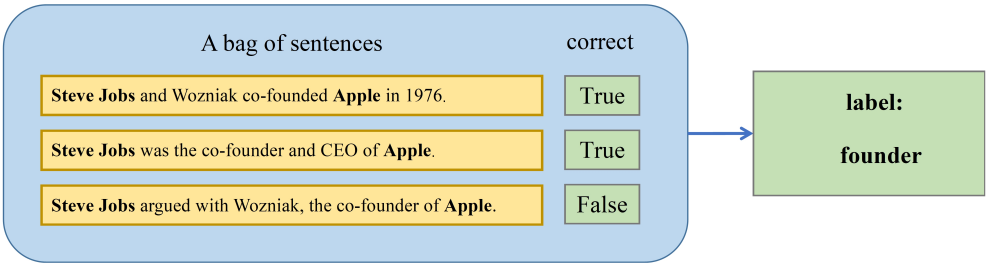

Figure 1. Example of a DS labeling process.

To mitigate the effect of noisy labels, Riedel et al. [2] proposed a multi-instance learning (MIL) framework to solve the data noise problem. Zeng et al. [3] proposed a segmented

convolutional neural network (PCNN) based on MIL. Lin et al. [4] introduced an attention mechanism based on PCNN and proposed the APCNN model. Liu et al. [5] used the bidirectional gated recurrent unit (Bi-GRU) to encode syntactic dependency trees. Yan et al. [6] proposed PLSTM-CNN (piecewise-LSTM convolutional neural network) encoder to encode sentences. There are also DSRE methods that attempt to enhance model effectiveness through word-level attention [7], the fusion of external knowledge [8], entity descriptions [9], relational phrases [8], and knowledge migration from pre-trained models [10]. Some researchers [11,12] have naturally transferred data-rich relational knowledge to long-tail relations using relational hierarchies. Deep learning language representations, such as those learned by Transformer [13] through language modeling [14], have been shown to obtain useful linguistic and syntactic properties of text through unsupervised pre-training alone.

The existing approach REDSandT [15] attempts to use BERT pre-trained models and relationships between labels and entities to provide instances and label embeddings for DSRE to enhance relation extraction (RE) and capture a wide set of relationships. However, the method has some limitations. It uses the BERT encoder for semantic modeling of word vectors, which is mainly used to obtain semantic information about words and is somewhat inadequate for processing text sequences, which can limit the effectiveness of denoising. Meanwhile, in terms of relation embeddings, the method uses the TransE model [16], which mainly considers the vector difference representation of the relationship, which is not sufficient for the feature representation of the overall entity pairs, resulting in the loss of feature information of some entities during the relation embeddings process. In addition, the method uses softmax to compute and optimize all the category labels, which may contain noisy categories and thus can be interfered with by the noisy categories during the normalization process, leading to high bias in the model during the fitting process and adding unnecessary overhead.

Responding to the experiences and shortcomings of the work described above, we propose the contextual information interaction and relation embeddings (CIRE) model, a novel DSRE model improved based on contextual information interaction and relation embeddings. Firstly, Bi-LSTM is added based on BERT to continue modeling text sequences by utilizing the error repair capability of the gating mechanism to filter and supplement the sequence information and enhance the contextual information interaction. Methods combining BERT and Bi-LSTM have been shown to perform well in other areas, such as sarcasm detection [17]. Meanwhile, we combine entity pairs and vector differences of entity pairs in the relation embeddings layer to improve the relation embeddings accuracy. Finally, we use sparse softmax as a classifier to improve the control of noisy categories by controlling the number of output categories.

The main contributions of this paper can be summarized as follows:

- This paper proposes a novel contextual information interaction and relation embeddings (CIRE) method for DSRE. We utilize BERT and Bi-LSTM to construct a neural network model. Based on acquiring semantic features based on the BERT encoder, we utilize the Bi-LSTM gating mechanism's error repair ability to filter and supplement the sequence information and further model the text sequence to extract more compact and rich semantic and sequence information.
- We improve the TransE model by building entity-guided augmented relations in a nonlinear layer in a high-dimensional space, which improves the long-tail problem by combining entity pairs and vector differences of entity pairs in the relation embeddings part to form a relation-embedded representation that helps to recognize a broader range of relations.
- We integrate CIRE-based sentence representations, relation representations, and sentence-weighted representations at the semantic fusion layer to produce the final enhanced sentence-level representation. Finally, we use sparse softmax as a classifier, improving the classifier's ability to control the noise categories by controlling the

number of output categories so that the model fitting process reduces the bias and effectively handles the noise category interference.
- By conducting a large number of experiments on the NYT2010 [2] dataset, we proved that our method is effective and reasonable.

2. Related Work

RE is fundamental in many natural language processing applications, such as knowledge graphs [16], machine quizzing [18,19], and knowledge graph construction. Traditional supervised learning RE methods [20,21], although with high accuracy and reliable model results, require many manually labeled datasets, and constructing such datasets is time consuming and labor intensive. In 2009, Mintz et al. [1] summarized the previous research and referred to the approach of Wu et al. [22], which proposed using a DS approach to construct a dataset for an RE task. To mitigate the negative effects of mislabeling, some studies [2,23,24] adopted a multi-instance learning paradigm, where sentences with the same entity pairs are put into a sentence packet, and a packet representation is learned. Miwa and Bansal et al. [25] introduced sequence and structural information from dependency trees into neural networks. In past studies, the tree representation may have needed to be processed and transformed to assess structural information changes in binary trees [26]. This idea of data representation and transformation can be shared in the study of relational extraction and the entropy of structural information of binary trees. Zhou et al. [27] combined an attention mechanism with a bidirectional long- and short-term memory network (Bi-LSTM). Wen et al. [28] proposed a novel gated PCNN, which takes into account the effects of entity pairs and sentence context on word encoding. Ye and Luo [29] combined multi-label learning with convolutional neural networks. Some researchers also use graph convolutional neural network models [30]. Recently, attention mechanisms have attracted increasing attention in the research field [31]. The attention mechanism became fashionable after it was first applied to recurrent neural network (RNN) models for image categorization by the Google Mind team [32]. Subsequently, the attention mechanism was used in a machine translation task [33], which marked the first application of the attention mechanism in the field of natural language processing (NLP). In 2017, the introduction of the Transformer model [13] by the Google Machine Translation team attracted widespread attention. Subsequently, various Transformer-based models have been widely used in NLP with state-of-the-art performance [34,35]. In our work, we use BERT, based on the Transformer model combined with Bi-LSTM, to extract sentence features efficiently, reduce some useless information by augmenting the data using sub-tree parse (STP), improve the relation embeddings layer, and perform information fusion at the semantic fusion layer, and finally classify them using sparse softmax.

3. Methodology

This section presents the details of our proposed model architecture and improvement approach. Figure 2 depicts the proposed sentence coding framework based on DSRE: Our model is first fed with a specific instance of RE to obtain a representation of the input, such as positional embeddings and the encoding of byte pairs of tokens. Then, using the BERT encoder for semantic modeling of word vectors, the output of BERT is fed into Bi-LSTM for further sequence modeling to enhance contextual information interaction. We improve the TransE model in high-dimensional space and shape the relation embeddings representing entity-to-entity distance in high-dimensional vector space. As in Figure 2, for a limited variety of entity pairs, the feature representation of the word vectors of the entity pairs is obtained only through the BERT hidden layer to prevent overfitting. After that, we utilize relational attention to emphasize sentences related to the underlying relations and further reduce sentence-level noise through weights. Finally, this generates a highly expressive sentence representation. Based on multi-instance learning, sentence representations of specific entity pairs aggregate in a bag to generate bag representations. In order to reduce the bias of the model during the fitting process, sparse softmax is used as a classifier, which

enhances the ability of the classifier to control the noise categories by limiting the number of output categories and computing only the non-zero elements in order to better adapt to the noisy categories. The following is a specific description of our method.

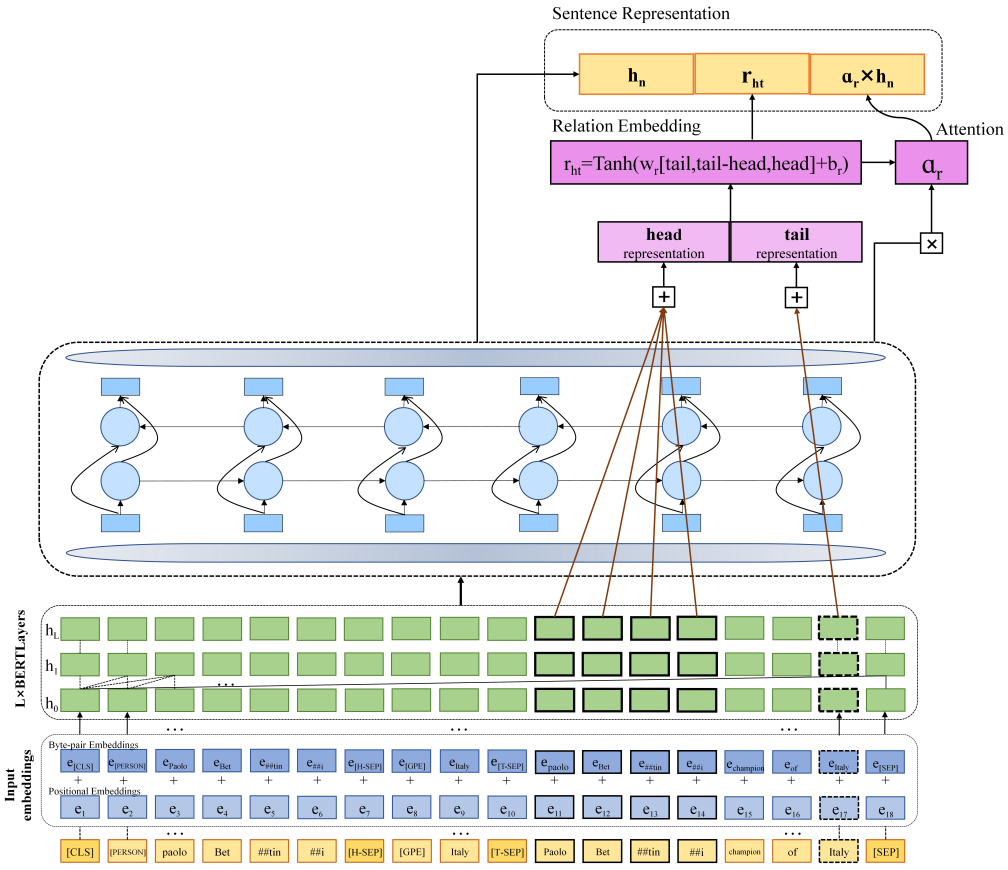

Figure 2. Sentence encoder for CIRE.

3.1. Sentence Input

Given a sentence *x* and a pair of entity pairs <head, tail>, CIRE constructs a distributed representation of the sentence by combining a relational attention-weighted sentence representation, a relation embedding representation, and sentence semantic features. Figure 3 below exemplifies the overall sentence input. The following sections describe the components of the model.

3.1.1. Input Representation

For each sentence's input representation, we encode it as a sequence of tokens. As shown in Figure 3, the tagging sequence of the model input contains three parts of information: the head entity type and tagging, the tail entity type and tagging, and the tagging sequence of the STP paths of the sentence. These three parts of information are split between them using the separators [H-SEP] and [T-SEP]. The head entity type and token, the tail entity type and token, and the token sequence of the STP path are the most critical information in the RE task. They are used to describe the information of entities, and the semantic information between entities, respectively, thus helping the model recognize the relationship between entities more accurately. In the BERT model, the beginning and the end of the input sequences are labeled using special token symbols, i.e., [CLS] and [SEP], to help the model process the different parts of the input sequences as well as the information related to the task. In addition, we compress the sentences into STP paths, as shown in Figure 3.

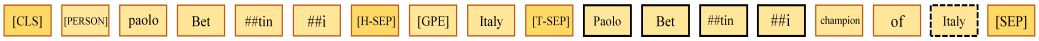

Figure 3. Sequence of sentence input markers.

Sub-tree parse of input sentences: To reduce the number of noisy words in the input sentences and to reduce the burden on the model to process long strings of text, we compress the original sentences into a sub-tree parse (STP) representation of the input sentences. STP preserves the sentence path connecting two entities to their least common ancestor (LCA) parent. This focuses attention on relational markup and shortening the length of the sentences. Using this approach improves the efficiency and accuracy of the model and reduces training time and resource consumption. At the same time, the STP path representation captures the relationships between phrases and clauses well and improves the generalization ability of the model so that it can handle input texts of different lengths and structures.

Entity type special marking: In an RE task, the type of an entity can constrain the relationship between entities. For example, in the sentence "Steve Jobs was the co-founder and CEO of Apple", "Steve Jobs" and "Apple" are regarded as entities whose types are "person" and "organization", respectively. Because entity types provide information about the concepts and attributes an entity represents, they can be used to constrain the relationships between entities. In the above sentence, the relationship "founder" can only exist between a person entity and an organization entity, not between two person entities or two organization entities. Therefore, entity type is regarded as a vital feature in RE, which can guide the model to extract and judge relationships. In the case where each relationship has type constraints on the participating entities [8,36], we include entity type information in the structured inputs of the model, as shown in Figure 3. Specifically, we combine 18 generic entity types that result from entity recognition using the spaCy model for NYT2010 sentences.

3.1.2. Input Embedding

The input embedding h_0 of BERT is a vector representation containing the entire structured input. This vector representation contains contextual information. Then, we input this vector representation with contextual information into the Bi-LSTM module for further sequence modeling of word vectors to capture the dependencies between words, thus improving the understanding of text sequences.

Byte pair tokens encoding: To utilize the subword information, we use byte pair encoding (BPE) to tag the input [37]. Byte pair encoding (BPE) is a commonly used algorithm for representing text as fixed-size words or subword units. This encoding method can effectively handle unregistered and low-frequency words, thus improving the model's performance. Our approach uses a tokenizer from a pre-trained model containing 30,000 tokens. To this, we added 20 specific tokens, such as [H-SEP], [T-SEP], and 18 entity type tokens to extend it. These additional tokens have special meanings in the input representation so that the tagger does not split them into subwords.

Position Coding: BERT learns a unique positional embedding for modeling the position of each word in a sentence. In this way, positional encoding can combine the positional information labeled for each input (subword) with semantic information to better capture contextual and semantic relations. Symmetry is the property of something that remains unchanged under some transformation. In the field of computers, symmetry is one of the important properties of the object of study. For relation extraction, relations can have different properties, including symmetry or asymmetry. Symmetry indicates that if there exists a relation such that A is related to B, then there also exists the same relation such that B is related to A. For example, a "brother" relationship is symmetric, while a "father–son" relationship is asymmetric. Models must be aware of role and positional differences between entities to predict relationships between different entity pairs correctly. Positional encoding, in this way, helps the model better understand the semantic information of the input sentences and more accurately represent the meaning of words in different contexts.

3.2. BERT

BERT (bidirectional encoder representations from Transformers) is a pre-training model with a multi-layer Transformer encoder that learns contextual relationships between words. The BERT encoder is unique because it is a bidirectional, non-directional pre-training model. In order to adapt to different application scenarios, BERT has several versions of pre-trained models to choose from. Among them, the two most commonly used versions are BERT-base and BERT-large.

Before using a pre-trained model, we obtained the relevant embeddings for each sentence by converting the input data into the appropriate input data format, as mentioned above. The detailed architecture of the BERT model is shown in Figure 4. E_N denotes the input embedding, *Trm* represents the Transformer encoder layer, and T_N is the hidden layer output vector.

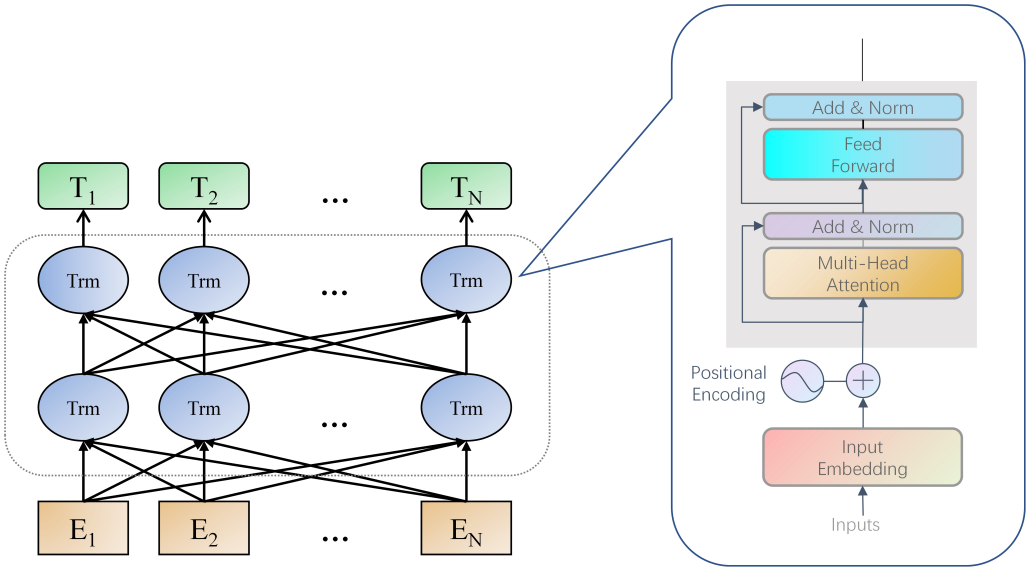

Figure 4. BERT encoder architecture.

3.3. Enhanced Text Message Processing

CIRE utilizes BERT and Bi-LSTM to construct neural network models. First, during the fine-tuning process, we used a structured input form specific to RE to minimize changes in the model architecture [38]. Then, we used a BERT encoder to semantically model the word vectors to obtain the word vector representations of the input sentences, which mainly acquired the semantic information of the words. We added the Bi-LSTM model in the sentence encoding session to further use the word vector information to model sequences and achieve contextual information interaction, which improves the model prediction accuracy. Meanwhile, the middle layer output of BERT may have redundancy and noise. Bi-LSTM utilizes the error repair capability of the gating mechanism to filter and supplement the text sequence information to extract more compact and rich semantic and sequence information.

3.4. Improved Constructed Sentence Representation Based on Relation Embeddings

Our approach uses a pre-trained BERT language model to convert the input sequence into a byte-level vector representation h_L as the initial features. We fine-tune the BERT model and use its output as the input to Bi-LSTM to further filter and supplement the sequence information to obtain the feature vectors of the sentence. Typically, in a BERT model, the [CLS] vector of the last layer is used as the output of the whole model [10] for downstream tasks such as input or classification operations. This is because the [CLS] vector is a summarized representation of the entire input sequence but does not contain the specific information needed to model each word in the sequence in detail. However, in our task, we

believe that not every word contributes the same importance to the feature representation of a sentence. Therefore, we chose to use the entire hidden layer containing the features of each word vector representation for the downstream task. Our core modules include entity attention, relation embedding, relational attention, and sentence representation. We investigate them in the following.

3.4.1. Acquisition of Head and Tail Entities

At the relation embeddings layer, we create head entity embeddings and tail entity embeddings by performing a summing operation on the vectors corresponding to the head entity and tail entity tokens in h_L of the last layer of the BERT model. Specifically, the BERT model encodes the input text as a series of token vectors, and each token vector contains the semantic information of that token in the context. By summing the token vectors where the head and tail entities are located, we can capture the contextual information of the head and tail entities into entity embeddings, which in turn can be used to compute relation embeddings.

We use head attention and tail attention to capture the relevant tokens of the head and tail entities. Through head attention, we assign a weight to each token to reflect its relevance to the head entity. Similarly, through tail attention, we assign a weight to each token to reflect its relevance to the tail entity. In this way, we can utilize these weights to compute the weighted average embedding of markers related to head and tail entities for subsequent RE tasks:

$$\alpha_{it}^{h} = \begin{cases} 1 & \text{if} \quad t = \text{head in STP tokens} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

$$\alpha_{it}^{t} = \begin{cases} 1 & \text{if} \quad t = \text{tail in STP tokens} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

T is the number of STP tokens, h_{it} is the head entity or tail entity individual word vectors, and the head entity and tail entity embeddings are formed by the expressions below:

$$h_i = \sum_{t=1}^{T} \alpha_{it}^{h} \cdot h_{it} \tag{3}$$

$$t_i = \sum_{t=1}^{T} \alpha_{it}^{t} \cdot h_{it} \tag{4}$$

3.4.2. Methods for Fusing Supplementary Entity Pair Features in the Relation Embeddings Layer

We use the TransE model [16] to fuse entity pair vectors to formulate relation embeddings. The TransE model treats the embedding of the underlying relation l as the distance (difference) between the h and t embeddings ($l_i = t_i - h_i$), on which we fuse the head entity and the tail entity to perform feature supplementation to represent the relation embeddings. The traditional TransE model uses the distance between vectors to measure the relationship between entities. Specifically, it uses the L_1 paradigm between the vectors of the head and tail entities' vectors to compute the distance between them. However, there is a problem with this approach, in that the relation embeddings generated through the difference between the head entity and the tail entity in the vector space lose some of the important features of the entities. Complete characterization of entity pairs is very important for RE to determine the relationship. Entity pair characterization includes information such as entity type, which helps to determine the relationship between entity pairs, for example, names of people, places, etc. These entity pairs can provide important clues about the type of relationship that may exist between the entities. Assuming that there is a relationship between the entity pairs (h,t), we fuse the difference features between the head and tail entity vectors and the entity pair vectors (denoted using "[]"), apply a linear transformation

to shape the relational embedding of each sentence i, and finally activate it through the Tanh layer to capture possible nonlinearities:

$$l_i = \text{Tanh}(w_l[h_i; t_i - h_i; t_i] + b_l) \tag{5}$$

where w_l is the underlying relationship weight matrix, and $b_l \in R^{d_t}$ is the bias vector, denoting that b_l is a vector of real numbers of dimension d_t. Relation embeddings are labeled l to indicate possible basic relationships between two entities, not necessarily real relationships. The embeddings of the head entity h_i and the tail entity t_i reflect the relevant markings of the entities, which we capture using entity attention.

3.4.3. Sentence Representation Based on Relational Attention

Our CIRE model is trained on compressing raw sentences into a representation of STP paths, which preserves as many tokens as possible that can express a relation, but to further reduce sentence-level noise, we emphasize sentence tokens associated with the underlying relation l_i by calculating the relational attention weights α_r. We define h_n as the feature vector of the overall output of the neural network model; h_{n_i} represents the feature vector of sentence i, and n in the set is the number of sentences in the package:

$$\alpha_r = \frac{\exp(h_{n_i} l_i)}{\sum_{j=1}^{n} \exp\left(h_{n_j} l_i\right)} \tag{6}$$

We then weight the final output vector h_n of the CIRE model with the relation embeddings:

$$h_n{}' = \sum_{t=1}^{T} \alpha_r \cdot h_{n_{it}} \tag{7}$$

3.4.4. Methods for Constructing Final Sentence Representations

In the RE task, the relationship's judgment needs to consider the semantic information in the input text. Fusing the vectors output from the hidden layer, the relationship vectors, and the weighted representation vectors in the high-dimensional feature fusion layer can increase the model's representational and generalization capabilities.

Ultimately, the sentence representation $s_i \in \Re^{d_h * 3}$ is constructed at the semantic fusion layer by fusing multiple features from relation vectors, weighted instance embeddings, and hidden layer output vectors:

$$s_i = [h_n; l_i; h'_n] \tag{8}$$

It has been proved by ablation experiments (presented later) that our proposed methods are effective.

3.5. Bag-Level Characterization

CIRE generates probability distributions over possible relationships for a bag of sentences s_1, s_2,..., s_n involving specific entity pairs. We cluster sentence representations with the same entity pairs in a single package, intended to reduce noise due to mislabeled information accompanying the data samples. However, not all sentence pair packet representations have the same importance. To further optimize the effectiveness of packet representation, we employ a selective attention mechanism [4]. This mechanism highlights sentences that are more relevant to the goal relation and, thus, better captures the characteristics of the relation.

Selective attention represents the packet as a weighted sum of each sentence, from which α_i is computed from the learned representation of each sentence's sentence representation with relation r:

$$B = \sum_{i=1}^{n} \alpha_i s_i \tag{9}$$

$$\alpha_i = \frac{\exp(s_i r)}{\sum_{j=1}^{n} \exp(s_j r)} \tag{10}$$

Finally, we input the packet representation B into a sparse softmax classifier to obtain the relationship's probability distribution. First, the input vector is exponentiated to obtain positive values. Then, the exponentiated vector is normalized to ensure that the elements of the output vector sum to one.

$$p(r) = SparseSoftmax(W_r \cdot B + b_r) \tag{11}$$

W_r means the relation weight matrix and $b_r \in \Re^{d_r}$ means the bias vector. Figure 5 illustrates the general architecture of the entire task.

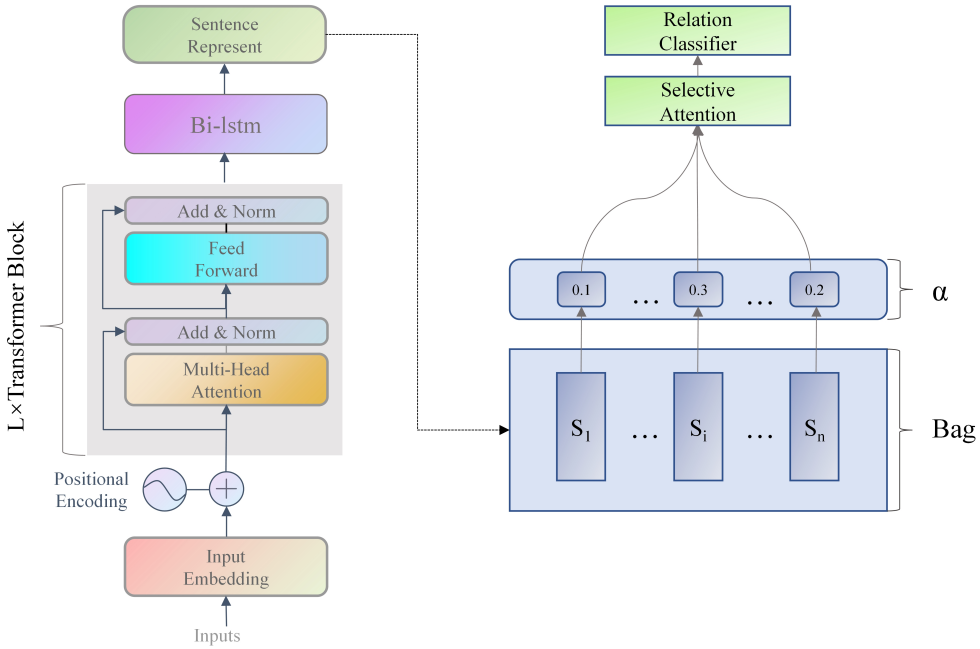

Figure 5. Overall model architecture.

3.6. Sparse Softmax Classifier

We use a dataset containing multiple relational categories in the DSRE practical application. The softmax classifier will compute and optimize all the labels, including the noise categories. As a result, the normalization will be interfered with by the noise categories, and high bias in the model fitting process occurs, which negatively affects the training and classification performance of the model and adds unnecessary overhead. Compared to softmax, we can better cope with the above situation by using sparse softmax, which improves the ability of the classifier to control the noise categories by controlling the number of output categories and calculating non-zero elements when performing model training and optimization, which results in less bias in the model fitting process. In addition, due to the small number of long-tailed relational samples, it is difficult for traditional softmax classifiers to accurately categorize the few categories in this case. In contrast, sparse softmax will tend to focus higher probability mass on a few categories, thus better capturing the features and patterns of the few categories. Table 1 shows the comparison between the sparse softmax and softmax formulas. Sparse softmax handles large-scale classification problems well and reduces computational and storage overheads.

Table 1. Comparison of softmax and sparse softmax formulas.

	Origin	Sparse
Softmax	$p_i = \frac{e^{s_i}}{\sum_{j=1}^{n} e^{s_j}}$	$p_i = \begin{cases} \frac{e^{s_i}}{\sum_{j \in \Omega_k} e^{s_j}}, i \in \Omega_k \\ 0, i \notin \Omega_k \end{cases}$
Cross-Entropy	$\log\left(\sum_{i=1}^{n} e^{s_i}\right) - s_t$	$\log\left(\sum_{i \in \Omega_k} e^{s_i}\right) - s_t$

4. Experiments

4.1. Datasets

Our experiments use an extensive benchmark DS dataset, the New York Times NYT, which we describe in detail below.

NYT: There are two different versions of this dataset, i.e., NYT2010 [2] and NYT-multi [39]. Taking NYT2010 as an example, we visualize the number of examples for the top 20 relationships that have the highest number of examples except for the NA relationship, and the situation is as shown in Figure 6, which contains the relationships between locations that have several times more examples than the other relationships have, and there is a serious long-tail phenomenon in these 20 relationships. In our experiments, the NYT dataset refers to NYT2010, which was constructed by aligning the triples in Freebase with the NYT corpus, and which has 53 relations, including the NA relation, indicating no relationship between these two entities. The training set contains 522,611 sentences, the test set contains 172,448 sentences, the training set contains 281,270 entity pairs, and the test set contains 96,678 entity pairs, as shown in Table 2. For ease of implementation, we provide an enhanced version of the dataset, NYT-2010enhanced, which includes syntax tree path (STP) and dependency tree path (SDP) versions of the input sentences and information on the type of the head and tail entities.

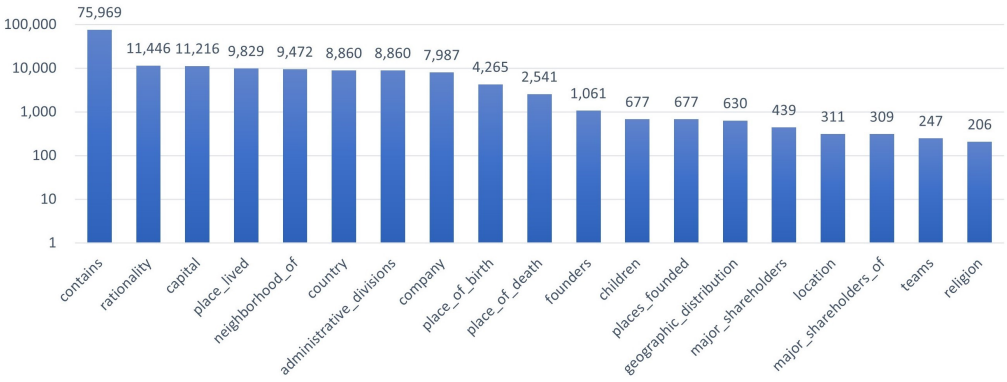

Figure 6. Distribution of selected data from the NYT2010 dataset.

Table 2. Data statistics for the NYT2010 dataset. "Ins", "EP", and "Rel" are number of instances, entity pairs, and number of relationships, respectively.

	Train	Test
Ins	522,611	172,448
EP	281,270	96,678
Rel	53	53

4.2. Hyperparameter Settings

We refer to Christou and Tsoumakas [15] for parameterizing our proposed model, and in our experiments, we use the BERT-base model, which has a hidden layer dimension $D_h = 768$ and sets the Max_Seq_Length $D_t = 64$. For the hyperparameters of the model, we adjust them manually in the course of the experiments. As in Table 3, we choose batch size = 32 in {8, 16, 32}. In {3, 4}, we choose epoch = 3. For the learning rate, we choose $lr = 2e^{-5}$ in $\{2e^{-4}, e^{-5}, 2e^{-5}\}$. In {0.2, 0.4, 0.5}, dropout p = 0.4. In {0.01, 0.001}, we choose weight decay = 0.001. In addition, we optimized the model using the Adam optimization scheme [40]. We minimize the loss using the cross-entropy criterion that weights the classes of the dataset to deal with the unbalanced training set. The experiments were conducted on a PC with 43 GB of RAM and 12 virtual central processing units (vCPUs) using an Intel(R) Xeon(R) Platinum 8255C CPU@2.50 GHz with an RTX 2080 Ti graphics card (11 GB). Also, the experiment used the Windows 10 operating system as the operating system software environment.

Table 3. Hyperparameter setting values.

Parameter Name	Value	Candidate Set
Max_seq_length	64	{32, 64, 128}
Batch size	32	{8, 16, 32, 64}
Epochs	3	{2, 3, 4, 5}
Learning rate	$2e^{-5}$	$\{2e^{-4}, e^{-5}, 2e^{-5}\}$
Dropout	0.4	{0.2, 0.4, 0.5}
Weight_decay	0.001	{0.01, 0.001}

4.3. Comparison Experiment

We conducted comparative experiments to compare our proposed approach with the advanced baseline approach on the NYT2010 dataset.

4.3.1. Advanced Baseline Model

Our proposed complex model CIRE is further compared with advanced baseline models. We compare the following models: (1) Mintz [1] is the first model proposed to solve the DSRE problem. (2) PCNN + ATT [4] uses a selective attention mechanism to mitigate the mislabeling problem. (3) RESIDE [8] uses graph convolutional networks (GCNs) for RE, which enhances RE by using external knowledge such as entity descriptions and entity types. (4) DISTRE [10] is a Transformer-based model that GPT fine-tuned for DSRE. (5) REDSandT [15] is a BERT-based RE model. It can handle long-tail relationships while reducing noise.

4.3.2. Evaluation Indicators

Our evaluation metrics include the precision–recall (PR) curve, the area under the curve (AUC), the Top-N precision (P@N), the mean (P@Mean), and the distribution of the relationships of the top 300 predictive relationships, as defined below.

The precision–recall (PR) curve, in which the horizontal axis represents the recall rate and the vertical axis represents the precision rate, shows the relationship between the precision rate and the recall rate at different thresholds.

The AUC is the area under the PR curve. A high AUC value indicates that the model performs better under different thresholds.

P@N is used to measure the accuracy of the model in the first N-predicted outcomes. By specifying different values of N, we can obtain the accuracy of the model in a range of different numbers of predictions.

P@Mean refers to the mean of the P@N values for different N values.

Depending on the specific prediction tasks and requirements, appropriate metrics can be selected to measure the model's accuracy, coverage, error rate, etc. Meanwhile, it may

be necessary to weigh the relationship between different metrics for different application scenarios to determine the most suitable model selection and optimization strategies. Some researchers propose a predictive statistics method applied to structure-based prediction models [41].

4.3.3. Analytical Comparisons with Benchmark Models

Figure 7 compares the CIRE and state-of-the-art model precision–recall curves, and we can observe:

(1) The neural network-based approach has a more significant advantage over probabilistic methods (Mintz) for information extraction. This is because neural networks can automatically learn and extract features from textual data without artificially designing feature templates to limit the model's performance.

(2) RESIDE, DISTRE, REDSandT, and CIRE outperform PCNN + ATT. PCNN + ATT has the highest accuracy rate initially but soon declines. This reveals the importance of entity types and relationship aliases as side information and transfer knowledge.

(3) RESIDE performs better at low recall because the model gives a lot of side information.

(4) Although DISTRE exhibits a precision value of 0.45 at medium recall, the precision is lower (2–12%) at recall < 0.25 compared to RESIDE, REDSandT, and CIRE.

(5) Overall, our model outperforms the baseline REDSandT, demonstrating the validity of our method.

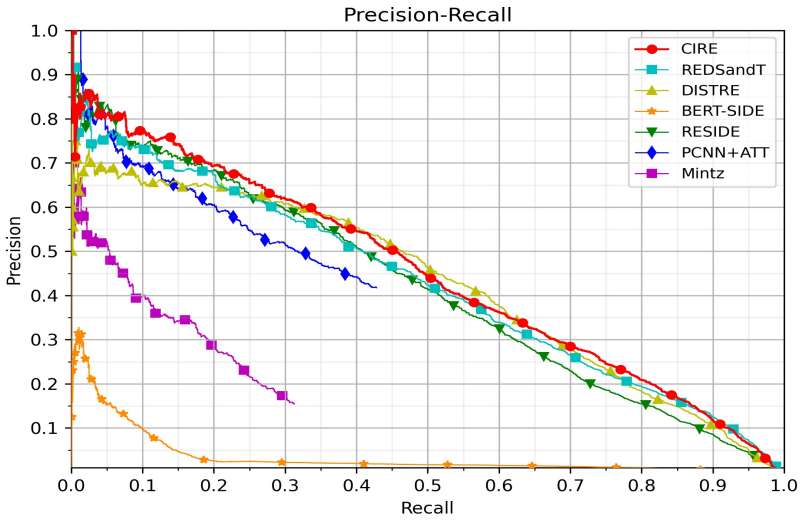

Figure 7. Precision vs. recall plot for CIRE on NYT2010 dataset.

Our model exhibits a steady decline, with the overall level above the REDSandT baseline. This means that the model improves the precision of the prediction results while maintaining a certain level of recall. This is because our proposed CIRE model adds Bi-LSTM to the REDSandT model, which in turn filters and complements the text sequence information to extract more compact and rich semantic and sequence information. We improve the relation embeddings part by fusing and supplementing the entity pairs' feature vectors. This utilizes the feature information of the entity pairs to enhance the relation embedding accuracy and helps identify a broader range of relationships. Finally, sparse softmax is used as a classifier to improve the classifier's ability to control the noise categories by controlling the number of output categories and calculating the non-zero elements, which reduces bias in the model fitting process. Compared to the baseline model, our method improves the overall recall level. The results of CIRE compared to other baseline methods for AUC, P@N, and P@Mean on the dataset NYT2010 are presented in Tables 4 and 5. These show that all our proposed CIRE models outperform the other baseline models.

Table 4. AUC values for CIRE and baseline model comparison.

DSRE Methods	AUC
Mintz	0.17
PCNN + ATT	0.341
RESIDE	0.415
DISTRE	0.422
REDSandT	0.424
CIRE (ours)	0.45

Table 5. Comparison of P@N and P@Mean values for CIRE and baseline models.

DSRE Methods	P@100	P@300	P@500	P@Mean
Mintz	52.3	45.0	39.7	45.67
PCNN + ATT	73	67.3	63.6	67.97
RESIDE	81.8	74.3	69.7	75.26
DISTRE	68	65.3	65	66.1
REDSandT	78	73	67.6	72.87
CIRE (ours)	79	77.7	71.2	75.97

Table 4 demonstrates a comparison of our model with the five baseline models. It can be seen that the AUC value of our model CIRE is improved by 2.6% compared to the baseline REDSandT model. The improvement in the AUC value implies that the model can better distinguish between right and wrong instances and reduces the over-reliance on mislabeling, proving our method's effectiveness. In Table 5, although the P@100 metric of our model is slightly lower than RESIDE, the small number of top 100 relationship instances and the low probability that these instances are noise do not highlight the overall level. The P@300 and P@500 values are significantly higher than the other baselines, and the average value P@Mean reached 75.97. The improvement in our model at P@300 and P@500 means that the model is better at ranking positive examples in the top 300 and top 500 predictions, and the model can rank true examples ahead of negative examples, resulting in a higher percentage of positive examples in the top 300 or top 500 predictions.

Figure 8 shows the distribution of the top 300 instance-predicted relationship types for CIRE and the other baseline models. Of the top 300 instance-predicted relationship types, CIRE recognizes 11 different relationship types, one of which (/person/religion) is not recognized by the other models. PCNN + ATT centrally favors a broad set of four relationship types, while RESIDE captures three additional types compared to PCNN + ATT. The fact that DISTRE and REDSandT can recognize more types than RESIDE and PCNN + ATT indicates the importance of knowledge transfer. Our model updates the relation embeddings based on knowledge transfer so that the relationships recognized by our model are not highly biased towards relationships with a high number of examples. It is proved that our model is effective in solving the long-tail problem.

4.4. Ablation Experiment

To evaluate the effectiveness of the different modular approaches of REDSandT, we performed ablation modeling studies on the NYT2010 dataset.

CIRE without Merge.ht: Remove the head and tail entities from the fusion layer in the improved TransE model.

CIRE without Bi-LSTM: Remove the Bi-LSTM module from CIRE.

CIRE without sparse softmax: Use softmax instead of sparse softmax in classification.

CIRE without h_n: Remove h_n from the feature fusion layer.

The experimental data in Tables 6 and 7 indicate that all of our modules contributed to the validity of our final model.

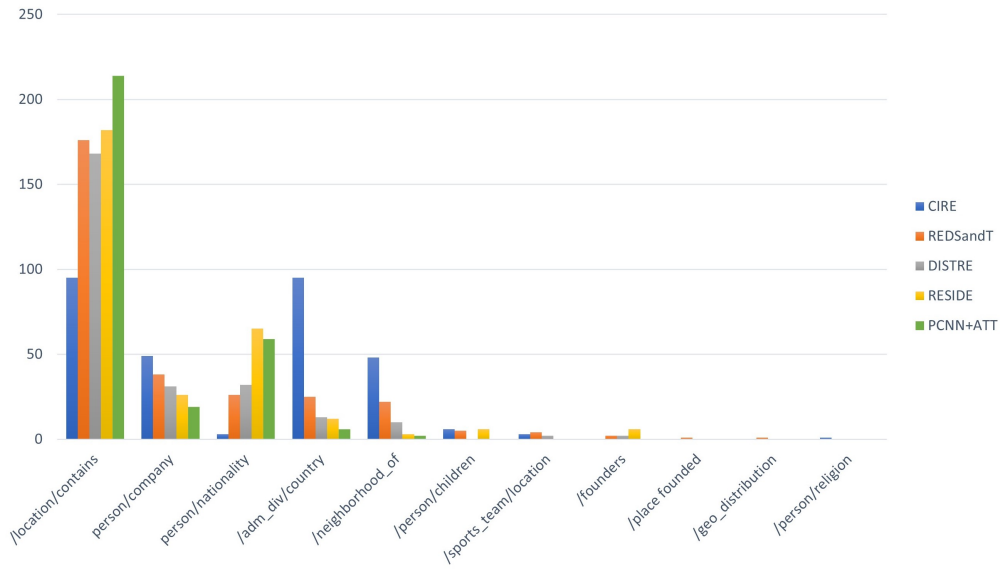

Figure 8. Distribution of predicted relationship types for the top 300 instances of CIRE and baseline models.

Table 6. AUC values for ablation experiments.

Metrics	AUC
CIRE without Merge.ht	0.429
CIRE without Bi-LSTM	0.435
CIRE without sparse softmax	0.436
CIRE without h_n	0.429
CIRE	0.45

Table 7. Ablation experiment P@N value vs. P@Mean value.

Metrics	P@100	P@300	P@500	P@Mean
CIRE without Merge.ht	77	70	68.4	71.8
CIRE without Bi-LSTM	76	74.7	70.2	73.63
CIRE without sparse softmax	76	74	69	73
CIRE without h_n	76	74	66.8	72.3
CIRE	79	77.7	71.2	75.97

Effectiveness of Bi-LSTM for sequence information complementation: Bi-LSTM uses the error-fixing ability of the gating mechanism to continue modeling on text sequences, which in turn filters and complements the text sequence information to extract more compact and rich semantic and sequence information. Therefore, removing the Bi-LSTM module will cause the model's performance to deteriorate. The comparison of the CIRE and CIRE without Bi-LSTM methods shows that the AUC value of CIRE without Bi-LSTM is reduced from 0.45 to 0.435 compared to the CIRE model. There is also a numerical decrease in the P@N value and the P@M value, proving that adding Bi-LSTM in the sentence encoding link effectively supplements the information of text sequences.

Effectiveness of entity pair feature supplementation at the relation embeddings layer: In the relation embeddings layer, the vectorial difference of the entity pairs represents the relationship based on which we fuse and supplement the feature vectors of the entity pairs themselves to improve the relation embedding accuracy. According to the comparison of CIRE and CIRE without Merge.ht methods, it can be seen that the AUC value of CIRE without Merge.ht reduces from 0.45 to 0.429 compared to the CIRE model, which is the most significant decrease and the biggest impact. The P@N value and P@Mean value are

also reduced, proving that the fusion and supplementation of entity pairs' features in the relation embeddings part is effective.

Effectiveness of sparse softmax classifier: CIRE uses sparse softmax as a classifier to improve the ability of the classifier to control the noise categories by controlling the number of output categories and calculating the non-zero elements, which leads to a reduction in bias in the model fitting process. Based on the comparison of CIRE and CIRE without sparse softmax methods, it can be seen that the AUC value of CIRE without sparse softmax is reduced from 0.45 to 0.436 as compared to the CIRE model. The P@N value and the P@Mean value are also reduced, proving that the sparse softmax classifier is effective in dealing with noise disturbances.

Effectiveness of fusion of sentence information output from the hidden layer: In constructing the feature fusion layer of the final sentence representation, fusing the sentence information output from the encoder hidden layer as the sub-information of the final sentence representation can enhance the model's ability to judge the relationship and improve the model's performance on unknown data, thus improving the accuracy of the relation extraction task. According to the comparison of the CIRE and CIRE without h_n methods, it can be seen that the AUC value of CIRE without h_n is reduced from 0.45 to 0.429 compared to the CIRE model. The P@N value and the P@Mean value are also reduced, proving that fusing the sentence information output from the encoder hidden layer in the feature fusion layer is effective.

It can be proved through experiments that this research has practical significance in applications. The DSRE task is of great practical significance in automated information extraction, large-scale relationship discovery, knowledge graph construction and application, and domain specialization application, which can help people extract valuable relationship information from large-scale textual data and apply it to various practical scenarios and applications.

5. Conclusions

In this paper, to solve the problems of a single method of text information processing, underutilization of feature information of entity pairs for relation embeddings, and mislabeling interference for multi-label classification in the DSRE domain, we propose a novel approach, CIRE, for DSRE. This first performs semantic modeling of word vectors using BERT and, then, continues to model textual sequence information by utilizing the error repair capability of Bi-LSTM's gating mechanism to extract more compact and rich semantic and sequence information. At the same time, we combine the entity pairs and the vector difference of entity pairs in the relation embeddings layer, which fuses and complements the features of entity pairs to improve the relation embedding accuracy. Finally, we choose sparse softmax as the classifier to improve the control of noise categories by controlling the number of output categories. After a large number of experimental results, our method is proved to be feasible and reasonable. On the public NYT2010 dataset, the AUC metric of our proposed method improves by 2.6% compared to the baseline method. Although our proposed method achieves outstanding results, there is still room for improvement in the long-tail problem. In the future, we propose better solutions for the long-tail and noise problems.

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

Funding: This research was funded by National Natural Science Foundation of China grant number 61966034.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: https://github.com/DespinaChristou/REDSandT (accessed on 20 November 2022).

Acknowledgments: The authors would like to thank the anonymous reviewers for their contribution to this paper.

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

References

1. Mintz, M.; Bills, S.; Snow, R.; Jurafsky, D. Distant supervision for relation extraction without labeled data. In Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, Singapore, 2–7 August 2009; pp. 1003–1011.
2. Riedel, S.; Yao, L.; McCallum, A. Modeling relations and their mentions without labeled text. In Proceedings of the Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2010, Barcelona, Spain, 20–24 September 2010; Proceedings, Part III 21, pp. 148–163.
3. Zeng, D.; Liu, K.; Chen, Y.; Zhao, J. Distant supervision for relation extraction via piecewise convolutional neural networks. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portuga, 17–21 September 2015; pp. 1753–1762.
4. Lin, Y.; Shen, S.; Liu, Z.; Luan, H.; Sun, M. Neural relation extraction with selective attention over instances. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Berlin, Germany, 7–12 August 2016; pp. 2124–2133.
5. Liu, T.; Zhang, X.; Zhou, W.; Jia, W. Neural relation extraction via inner-sentence noise reduction and transfer learning. *arXiv* **2018**, arXiv:1808.06738.
6. Yan, D.; Hu, B. Shared representation generator for relation extraction with piecewise-lstm convolutional neural networks. *IEEE Access* **2019**, *7*, 31672–31680. [CrossRef]
7. He, Z.; Chen, W.; Li, Z.; Zhang, M.; Zhang, W.; Zhang, M. See: Syntax-aware entity embedding for neural relation extraction. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32.
8. Vashishth, S.; Joshi, R.; Prayaga, S.S.; Bhattacharyya, C.; Talukdar, P. Reside: Improving distantly-supervised neural relation extraction using side information. *arXiv* **2018**, arXiv:1812.04361.
9. Hu, L.; Zhang, L.; Shi, C.; Nie, L.; Guan, W.; Yang, C. Improving distantly-supervised relation extraction with joint label embedding. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 3–7 November 2019; pp. 3821–3829.
10. Alt, C.; Hübner, M.; Hennig, L. Fine-tuning pre-trained transformer language models to distantly supervised relation extraction. *arXiv* **2019**, arXiv:1906.08646.
11. Han, X.; Yu, P.; Liu, Z.; Sun, M.; Li, P. Hierarchical relation extraction with coarse-to-fine grained attention. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 2236–2245.
12. Yu, E.; Han, W.; Tian, Y.; Chang, Y. Tohre: A top-down classification strategy with hierarchical bag representation for distantly supervised relation extraction. In Proceedings of the 28th International Conference on Computational Linguistics, Online, 8–13 December 2020; pp. 1665–1676.
13. 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, Long Beach, CA, USA, 4–9 December 2017; Volume 30.
14. Verga, P.; Strubell, E.; McCallum, A. Simultaneously self-attending to all mentions for full-abstract biological relation extraction. *arXiv* **2018**, arXiv:1802.10569.
15. Christou, D.; Tsoumakas, G. Improving distantly-supervised relation extraction through bert-based label and instance embeddings. *IEEE Access* **2021**, *9*, 62574–62582. [CrossRef]
16. Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; Yakhnenko, O. Translating embeddings for modeling multi-relational data. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 5–8 December 2013; Volume 26.
17. Viji, D.; Revathy, S. A hybrid approach of Weighted Fine-Tuned BERT extraction with deep Siamese Bi–LSTM model for semantic text similarity identification. *Multimed. Tools Appl.* **2022**, *81*, 6131–6157. [CrossRef]
18. Yao, X.; Van Durme, B. Information extraction over structured data: Question answering with freebase. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Baltimore, MD, USA, 22–27 June 2014; pp. 956–966.
19. Yu, M.; Yin, W.; Hasan, K.S.; dos Santos, C.; Xiang, B.; Zhou, B. Improved neural relation detection for knowledge base question answering. *arXiv* **2017**, arXiv:1704.06194.
20. Zelenko, D.; Aone, C.; Richardella, A. Kernel methods for relation extraction. *J. Mach. Learn. Res.* **2003**, *3*, 1083–1106.
21. Culotta, A.; Sorensen, J. Dependency tree kernels for relation extraction. In Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04), Barcelona, Spain, 21–26 July 2004; pp. 423–429.

22. Wu, F.; Weld, D.S. Autonomously semantifying wikipedia. In Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management, Lisbon, Portugal, 6–10 November 2007; pp. 41–50.
23. Hoffmann, R.; Zhang, C.; Ling, X.; Zettlemoyer, L.; Weld, D.S. Knowledge-based weak supervision for information extraction of overlapping relations. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Portland, OR, USA, 19–24 June 2011; pp. 541–550.
24. 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.
25. Miwa, M.; Bansal, M. End-to-end relation extraction using lstms on sequences and tree structures. *arXiv* **2016**, arXiv:1601.00770.
26. Jäntschi, L.; Bolboacă, S.D. Informational entropy of B-ary trees after a vertex cut. *Entropy* **2008**, *10*, 576–588. [CrossRef]
27. Zhou, P.; Shi, W.; Tian, J.; Qi, Z.; Li, B.; Hao, H.; Xu, B. Attention-based bidirectional long short-term memory networks for relation classification. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Berlin, Germany, 7–12 August 2016; pp. 207–212.
28. Wen, H.; Zhu, X.; Zhang, L.; Li, F. A gated piecewise CNN with entity-aware enhancement for distantly supervised relation extraction. *Inf. Process. Manag.* **2020**, *57*, 102373. [CrossRef]
29. Ye, H.; Luo, Z. Deep ranking based cost-sensitive multi-label learning for distant supervision relation extraction. *Inf. Process. Manag.* **2020**, *57*, 102096. [CrossRef]
30. Xu, J.; Chen, Y.; Qin, Y.; Huang, R.; Zheng, Q. A feature combination-based graph convolutional neural network model for relation extraction. *Symmetry* **2021**, *13*, 1458. [CrossRef]
31. Chaudhari, S.; Mithal, V.; Polatkan, G.; Ramanath, R. An attentive survey of attention models. *ACM Trans. Intell. Syst. Technol. (TIST)* **2021**, *12*, 1–32. [CrossRef]
32. Mnih, V.; Heess, N.; Graves, A. Recurrent models of visual attention. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; Volume 27.
33. Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv* **2014**, arXiv:1409.0473.
34. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
35. Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R.R.; Le, Q.V. Xlnet: Generalized autoregressive pretraining for language understanding. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; Volume 32.
36. Liu, Y.; Liu, K.; Xu, L.; Zhao, J. Exploring fine-grained entity type constraints for distantly supervised relation extraction. In Proceedings of the Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, Dublin, Ireland, 23–29 August 2014; pp. 2107–2116.
37. Sennrich, R.; Haddow, B.; Birch, A. Neural machine translation of rare words with subword units. *arXiv* **2015**, arXiv:1508.07909.
38. Dong, L.; Yang, N.; Wang, W.; Wei, F.; Liu, X.; Wang, Y.; Gao, J.; Zhou, M.; Hon, H.W. Unified language model pre-training for natural language understanding and generation. *Adv. Neural Inf. Process. Syst.* **2019**, *32*.
39. Cabot, P.L.H.; Navigli, R. REBEL: Relation extraction by end-to-end language generation. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2021, Online, 7–11 November 2021; pp. 2370–2381.
40. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
41. Bolboacă, S.D.; Jäntschi, L. Predictivity approach for quantitative structure-property models. Application for blood-brain barrier permeation of diverse drug-like compounds. *Int. J. Mol. Sci.* **2011**, *12*, 4348–4364. [CrossRef] [PubMed]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.