

## Article

# Leveraging Prompt and Top-K Predictions with ChatGPT Data Augmentation for Improved Relation Extraction

Ping Feng <sup>1,2,3,4,5</sup> , Hang Wu <sup>6</sup>, Ziqian Yang <sup>6</sup>, Yunyi Wang <sup>6</sup> and Dantong Ouyang <sup>1,\*</sup> 

<sup>1</sup> College of Computer Science and Technology, Jilin University, Changchun 130012, China; fengping@ccu.edu.cn

<sup>2</sup> College of Computer Science and Technology, Changchun University, Changchun 130022, China

<sup>3</sup> Ministry of Education Key Laboratory of Intelligent Rehabilitation and Barrier-Free Access for the Disabled, Changchun 130022, China

<sup>4</sup> Jilin Provincial Key Laboratory of Human Health State Identification and Function Enhancement, Changchun 130022, China

<sup>5</sup> Jilin Rehabilitation Equipment and Technology Engineering Research Center for the Disabled, Changchun 130022, China

<sup>6</sup> College of Cybersecurity, Changchun University, Changchun 130022, China; wuhang@163.com (H.W.); 19003063186@163.com (Z.Y.); 17304352555@163.com (Y.W.)

\* Correspondence: dantong1206@163.com; Tel.: +86-13504313045

**Abstract:** Relation extraction tasks aim to predict the type of relationship between two entities from a given text. However, many existing methods fail to fully utilize the semantic information and the probability distribution of the output of pre-trained language models, and existing data augmentation approaches for natural language processing (NLP) may introduce errors. To address this issue, we propose a method that introduces prompt information and Top-K prediction sets and utilizes ChatGPT for data augmentation to improve relational classification model performance. First, we add prompt information before each sample and encode the modified samples by pre-training the language model RoBERTa and using these feature vectors to obtain the Top-K prediction set. We add a multi-attention mechanism to link the Top-K prediction set with the prompt information. We then reduce the possibility of introducing noise by bootstrapping ChatGPT so that it can better perform the data augmentation task and reduce subsequent unnecessary operations. Finally, we investigate the predefined relationship categories in the SemEval 2010 Task 8 dataset and the prediction results of the model and propose an entity location prediction task designed to assist the model in accurately determining the relative locations between entities. Experimental results indicate that our model achieves high results on the SemEval 2010 Task 8 dataset.

**Keywords:** relation extraction; language model; prompt information; deep learning models; NLP



**Citation:** Feng, P.; Wu, H.; Yang, Z.; Wang, Y.; Ouyang, D. Leveraging Prompt and Top-K Predictions with ChatGPT Data Augmentation for Improved Relation Extraction. *Appl. Sci.* **2023**, *13*, 12746. <https://doi.org/10.3390/app132312746>

Academic Editors: Julian Szymanski and Ahmed Rafea

Received: 4 November 2023

Revised: 21 November 2023

Accepted: 23 November 2023

Published: 28 November 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

Relation extraction has garnered significant attention from researchers, as it is an important subtask in the information extraction task and plays an important role in many downstream natural language processing applications, e.g., sentiment analysis, question answering application, abstract summarization, and knowledge graph construction. If the entity pairs  $e_1$  and  $e_2$  are labeled and the types of relationships are predefined, the task becomes a standard classification problem. For example, “Bob Parks made a similar  $\langle e_1 \rangle$  offer  $\langle /e_1 \rangle$  in a  $\langle e_2 \rangle$  phone call  $\langle /e_2 \rangle$  made earlier this week.” In this text, “offer” is the head entity  $e_1$ , “phone call” is the tail entity  $e_2$ , and the relationship between the two entities is of type “Message-Topic ( $e_2, e_1$ )”.

With the emergence of language models, such as BERT [1], RoBERTa [2], GPT [3], etc., their powerful ability to capture contextual information has been proven. Currently, there is a significant amount of work in the field of relational extraction that involves initial fine-tuning of pre-trained language models to obtain vector representations containing rich

semantic information. Subsequently, researchers make algorithmic improvements on top of these models. In [4], it is demonstrated that a difference exists in the objective forms of the pre-trained language model (PLM) between the pre-training and fine-tuning phases. Typically, PLMs are usually pre-trained with a cloze-style task. However, in the fine-tuning phase, the vector representation of just one or a few words in the model may be utilized for a specific task. For example, when using the BERT model to do sentiment analysis on the sentence “This is a great movie”. First of all, a “[CLS]” special token is usually added in front of the sentence, which becomes “[CLS] This is a great movie”. Then, the vector representation of this sentence is obtained by BERT as  $X = \{x_0, x_1, x_2, \dots, x_t\}$ , where  $x_0$  is the representation of the “[CLS]” token. Next, a straightforward classification task uses only  $x_0$  through a linear layer. This approach may result in a model that only partially leverages the semantic information in the pre-trained language model.

For supervised relation extraction models, the quantity and quality of data in the training set significantly impact the final performance of the model. Currently, the training data for such models mainly relies on manual labeling to complete. While manually labeled data is generally reliable, labeling data is time-consuming and labor-intensive. Additionally, obtaining diverse expressions that represent the same semantic information from manually labeled data can be challenging. One of the purposes of data augmentation is to increase the diversity of training data, which can effectively alleviate the problem of data scarcity.

When using a pre-trained language model for relation extraction tasks, the final prediction is typically the label with the highest probability from the output probability distribution of the classification layer. Sometimes, the model incorrectly predicts relationships between entities, but the correct labels may be in the first K probability distributions, which contain valuable information for the relationship extraction task. However, it has yet to receive extensive attention for relational classification tasks. Some datasets define relationship categories by distinguishing the relative positions of entities, e.g., “Entity-Origin (e1, e2)” and “Entity-Origin (e2, e1)” are two different relationship categories.

This paper proposes a relation extraction method based on prompt information and Top-K prediction sets to address the above problems. Firstly, adding prompt information before each input data can link the pre-training and fine-tuning phases of the pre-trained language model. The prompt information and Top-K prediction set are effectively fused through the multi-head attention [5] to more fully utilize the rich semantic information of vectors. We added an entity position prediction method to assist the model in correctly predicting the relative positions of two entities. Additionally, augmented data generated based on ChatGPT improves the model’s generalization. We conducted corresponding experiments on the SemEval 2010 Task 8 dataset, and the results demonstrate that the method proposed in this paper significantly outperforms the baseline model in terms of F1 score.

The rest of the paper is organized as follows. Section 2 reviews previous work on relation extraction and prompt tuning. In Section 3, we present the details of the proposed method in this paper. Section 4 presents the dataset used, experimental steps, and experimental results. Finally, we show the conclusions of this paper and the prospects for future work in Section 5.

## 2. Related Work

Relation extraction is a crucial aspect of the NLP domain, aiming to determine the relationship between two entities in a given sentence. The performance of traditional relation extraction models depends on the quality of extracted features. However, feature extraction using NLP tools often introduces noise. An example is the ambiguity of word meanings. Consider a sentence: “I bought an apple”. In this context, “apple” could refer to a fruit or a technology company, and this ambiguity is a potential source of noise, which can degrade model performance. To reduce the noise introduced during feature extraction, in recent years, several models have emerged that use deep neural networks for supervised relation extraction tasks, which are capable of making predictions about the type of relationship

between specified entities in text. Ref. [6] introduced a model that employs Convolutional Neural Networks (CNNs) to extract lexical and sentence-level features. In this model, they initially convert each word into a vector through word embedding. Then, lexical-level features are extracted based on the given noun. Simultaneously, a CNN is employed to extract the sentence-level features. Finally, the two levels of features are fused into a final feature vector and fed into a SoftMax layer to predict the relationship between the two entities. Nevertheless, due to CNN's limitations, this model may require assistance in accurately predicting relationship types between entities, particularly in long sentences with entities positioned far apart. Graph Convolutional Networks (GCNs) [7] are a widely used structure in which the information of each node in each GCN layer communicates with neighboring nodes through edges between them. The ability of GCNs to efficiently capture semantic relationships and contextual information between entities in a text has been demonstrated by many previous studies [8–10]. Many models use dependency trees to build graphs. However, the graphs generated by this method can be noisy, particularly when generated automatically. Noise may be present in the generated graphs because the algorithms constructing the dependency trees cannot handle complex syntactic structures or the text is ambiguous. Excessive reliance on dependency trees may harm the performance of relational extraction tasks.

In recent years, pre-trained language models have gained significant attention in various research areas for their potent semantic representations. BERT belongs to the Transformer architecture family and stands out for its ability to perform bidirectional context modeling. Unlike traditional models, BERT considers both a word's left and right context, providing a more comprehensive understanding of language context. Trained unsupervised on large-scale unlabeled text data, BERT learns universal language representations, making it a versatile tool for various NLP tasks. The model's impact extends to tasks such as text classification, named entity recognition, and relation extraction, showcasing its effectiveness through fine-tuning. RoBERTa builds upon the foundation of BERT. Unlike BERT, RoBERTa removes the bidirectional training restriction and employs larger text corpora for pretraining, enhancing its language representation capabilities. Notable improvements include the introduction of dynamic masking, where the mask length is adjusted dynamically during each training iteration, facilitating better contextual learning. Ref. [11] proposed the R-BERT model, which uses BERT to extract relational features and fuses the information of head and tail entities to accomplish the relational extraction task, leading to a notable enhancement in model performance.

Using BERT for the relationship classification task inevitably brings a gap between pre-training and fine-tuning for traditional data pre-processing, impacting the model's performance [12]. To address this issue, a new fine-tuning paradigm, Prompt tuning, is proposed based on pre-trained language models. By utilizing language prompts as contextual cues, downstream tasks can be formulated as objectives akin to pre-training objectives. The addition of templates avoids the introduction of extra parameters, thus allowing the language model to achieve the desired results in small- or zero-sample scenarios. Large-scale models are believed to maximize their reasoning and comprehension capabilities with suitable templates. Ref. [13] introduced the proposed framework of rule-based prompt tuning. The method initially encodes the prior task knowledge into rules, breaks down the task into sub-tasks, proceeds to design the requisite sub-prompts, and ultimately assembles these sub-prompts to process the task by the established rules. This approach effectively narrows the gap between pre-training and fine-tuning and alleviates the challenges of designing prompt templates and sets of label words. Ref. [14] proposes a relational extraction method that adds prompt information and feature reuse. Firstly, the prompt information is added before each sentence. Then, the pre-trained language model RoBERTa encodes the sentence, entity pair, and prompt information. BIGRU is introduced into the composition of the neural network to extract the information. The feature information is passed through the neural network to form several sets of feature vectors. Then, these feature vectors are reused in different combinations to form multiple outputs. The outputs are aggregated

using the ensemble-learning soft voting for relational extraction.

In order to increase the amount of data in the training set to improve the performance of supervised models, ref. [15] three neural network machine translation systems were used to generate augmented data by back-translating to the original data. However, there is no way to pass the annotation information of the entities during the back-translation process, and it is necessary to add the entity alignment operation after the back-translation operation, which may result in the accumulation of errors and thus damage the model performance. Previous research in [16] demonstrates that when the model makes an incorrect prediction, it often finds the correct result among the top K labels with the highest probabilities, referred to as the Top-K prediction set. The Top-K prediction set contains valuable information for establishing connections between ground truth labels and other labels, which is beneficial for relational classification tasks.

### 3. Relational Extraction Model PTKRE

This paper proposes the PTKRE (Prompt and Top-K Relationship Extraction) model. The structure of the PTKRE model is shown in Figure 1, consisting primarily of four components: input layer, Top-K prediction set generation layer, multi-head attention layer, and entity location prediction layer. Firstly, the input layer converts the sentence with the prompt information into a vector representation  $H$ . We generate the Top-K prediction set and calculate the loss based on the vector representation  $H$ . Next, the multi-head attention mechanism fuses the Top-K prediction set and the two “<mask>” tokens in the vector representation  $H$ , where the <mask> tokens masked the relational category words. We then feed the fused vectors into the fully connected layer and utilize the loss function layer to calculate the loss. Meanwhile, use the “<s>” token and the two entities to form the three nodes in the graph, where the “<s>” token is a special token that needs to be added at the top of the sentence when using RoBERTa. Then, use the graph convolutional neural network for feature extraction and the feature vector to complete the entity location prediction tasks and get the loss.

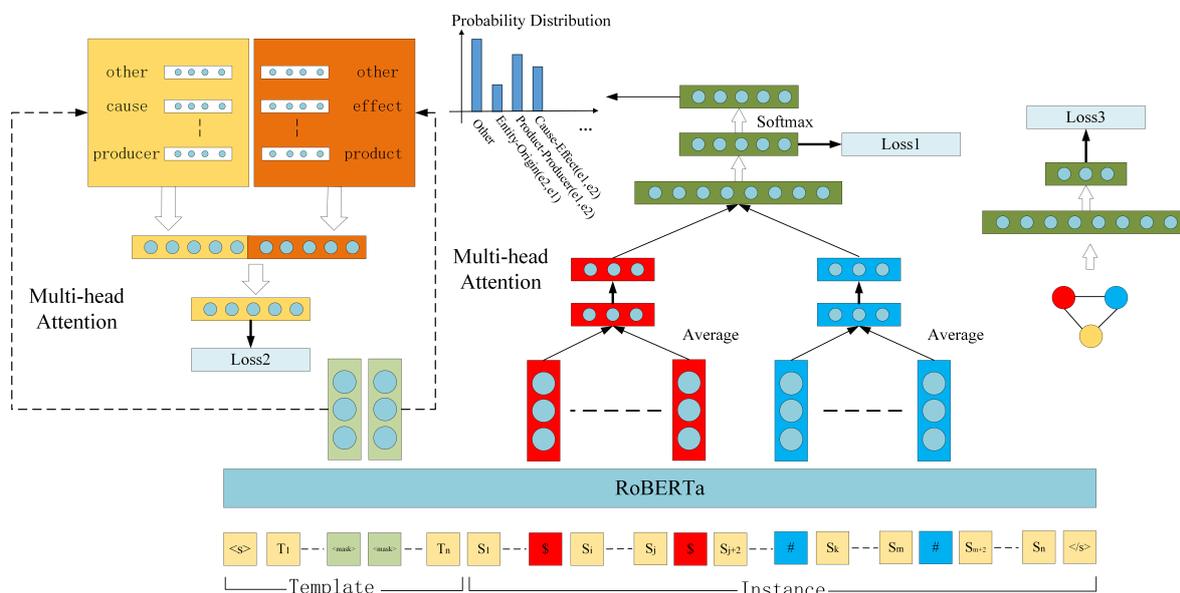


Figure 1. Overall structure of the model. In the figure, “\$” and “#” are special tokens, where  $S_1$  to  $S_j$  are head entities and  $S_k$  to  $S_m$  are tail entities.

#### 3.1. Input Layer

First, we introduce prompt information in the form of “\$ [ent1] \$ and # [ent2] # are related in the sentence through <mask> <mask> sentence”: at the beginning of the sentence. Subsequently, we replace “[ent1]” and “[ent2]” with the head entity and the tail entity in

the text, respectively. Specific examples are provided in Table 1. Next, we need to apply a replacement operation to all sentences in the dataset. The special token “<e1>” and “</e1>” denoting the start and end positions of the head entity in the sentences are replaced with the token “\$”, while the token “<e2>” and “</e2>” indicating the start and end positions of the tail entity are replaced with the token “#”. Finally, “<s>” is added at the beginning of each sentence, and “</s>” is added at the end. For example, a sentence in the dataset “A <e1> girl </e1> plays her <e2> violin </e2> on a pogo stick”. after processing would become: “<s> \$ girl \$ and # violin # are related in the sentence through <mask> <mask> sentence: A \$ girl \$ plays her # violin # on a pogo stick </s>”. Next, we use the previously modified sentence as the input sequence  $S_{input} = \{x_0, x_1, x_2, \dots, x_t\}$  for RoBERTa, and we finally obtain the corresponding vector representation  $H = \{h_0, h_1, h_2, \dots, h_t\}$  from RoBERTa, with  $h_0$  containing the feature information of the entire sentence.

**Table 1.** Add a prompt before a sentence.

Class	Example
before modification	A <e1> girl </e1> plays her <e2> violin </e2> on a pogo stick .
after modification	<s> \$ girl \$ and # violin # are related in the sentence through <mask> <mask> sentence: A <e1> girl </e1> plays her <e2> violin </e2> on a pogo stick. </s>

### 3.2. Top-K Prediction Set Generation Layer

The main purpose of this layer is to generate Top-K prediction sets for all samples. When provided with a vector representation  $H$  for a sentence, as generated by the input layer, we first extract the vectors associated with the “<s>” token and the vectors of the corresponding positions of the two entities in  $H$ . Since the sentences do not have fixed lengths for the head entity and the tail entity, performing separate average pooling on the representations of the head entity and the tail entity becomes necessary. The “<s>” token has a fixed length, requiring no additional processing. Then, the “<s>” token and the vector representations of the processed two entities are passed into the fully connected layer for processing, respectively, to obtain the vector representations  $s$ ,  $e_1$  and  $e_2$ . Here,  $s$ ,  $e_1$  and  $e_2 \in R^d$  and  $d$  are the size of the hidden layer vectors outputted by the pre-trained language model. For example, the RoBERTa-large model uses a dimension of 1024. The vector representations  $s$ ,  $e_1$ , and  $e_2$  are then concatenated:

$$r = FC([s \parallel e_1 \parallel e_2]) \quad (1)$$

The  $\parallel$  represents vector concatenation.  $FC$  is the fully connected layer. The representation  $r$  is input into the SoftMax and the loss function layers. The SoftMax layer calculates the probabilities of potential relationships between entities, and we select the Top-K most probable relationship categories by configuring the hyperparameter  $k$ , resulting in the formation of the Top-K prediction set. The cross-entropy loss is used in the loss function layer to optimize the model, and this loss is denoted as  $Loss_1$ .

### 3.3. Multi-Head Attention Layer

After obtaining the Top-K prediction set from the Top-K prediction set generation layer, it is first necessary to split each relational category word in the Top-K prediction set into two words. For example, “Instrument-Agency (e1, e2)” is split into two words, “Instrument” and “Agency”, and if “Instrument-Agency (e2, e1)” is split into two words, “Instrument” and “Agency”. The order of the split words is related to the order of “e1” and “e2” in the relational category words. Following this, we employ multi-attention to combine the relational representation with the representation of the “<mask>” token:

$$r_1 = MultiHeadAtt(M_h, L_h, L_h) \quad (2)$$

$$r_2 = \text{MultiHeadAtt}(M_t, L_t, L_t) \quad (3)$$

where  $L_h$  and  $L_t \in R^{k*d}$  are vector representations obtained by splitting the relationship category words, they are generated using RoBERTa.  $M_h$  and  $M_t$  are vector representations corresponding to the two "<mask>" tokens extracted from the vector representation  $H$ .  $M_h$  and  $L_h$  form a pair, with  $M_h$  as the query vector, resulting in vector  $r_1$  after passing through the multi-head attention layer.  $M_t$  and  $L_t$  also form a pair, with  $M_t$  as the query vector, resulting in vector  $r_2$  after passing through the multi-head attention layer. Finally, it is necessary to concatenate the vectors  $r_1$  and  $r_2$  and then use the Multilayer Perceptron (MLP) with a softmax activation function to obtain the final prediction:

$$p = \text{MLP}(r_1 \parallel r_2). \quad (4)$$

Optimize the model using cross-entropy loss function to obtain  $Loss_2$ .

### 3.4. Entity Location Prediction Layer

In the SemEval-2010 Task 8 dataset, relationship categories, except for "other", are corresponded in pairs, such as "Cause-Effect (e1, e2)" and "Cause-Effect (e2, e1)". Hence, distinguishing the relative positions of two entities in this dataset becomes crucial. To assist the model in the final relation extraction task, we first designed three categories: "head entity-tail entity", "tail entity-head entity", and "other", as shown in Table 2. We utilized the vector representations of the "<s>" token, the head entity, and the tail entity to construct nodes in an undirected graph. Edges link every pair of nodes among the three, and each node has self-connections. We then applied GCN to capture the topological features of the graph by computing new representations for each node. For multi-layer GCNs, the propagation rule is as follows:

$$H^{(l+1)} = \sigma(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^{(l)} W^{(l)}) \quad (5)$$

where  $\tilde{A} = A + I_N$  is the graph's adjacency matrix with the addition of self-connections for nodes, and where  $I_N$  is the identity matrix.  $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$  and  $W^{(l)}$  is a learnable weight matrix.  $\sigma(\cdot)$  represents the activation function. Finally, the updated features of the three nodes in the graph are spliced and used for the final prediction of the relative positions of the head and tail entities, and we obtain  $Loss_3$  using the cross-entropy loss function.

**Table 2.** The meaning of categories in entity position prediction task.

Class	Meaning
0	(e1, e2)
1	(e2, e1)
2	other

### 3.5. Data Augmentation

Based on the number of sentences for each relationship type in the SemEval 2010 Task 8 dataset, it is evident that the dataset suffers from imbalance. Consequently, we deliberately tried to perform more data augmentation on the categories with fewer sentences. For each sentence requiring augmentation, ChatGPT generated three additional sentences. We present the resulting data augmentation quantities in Table 3.

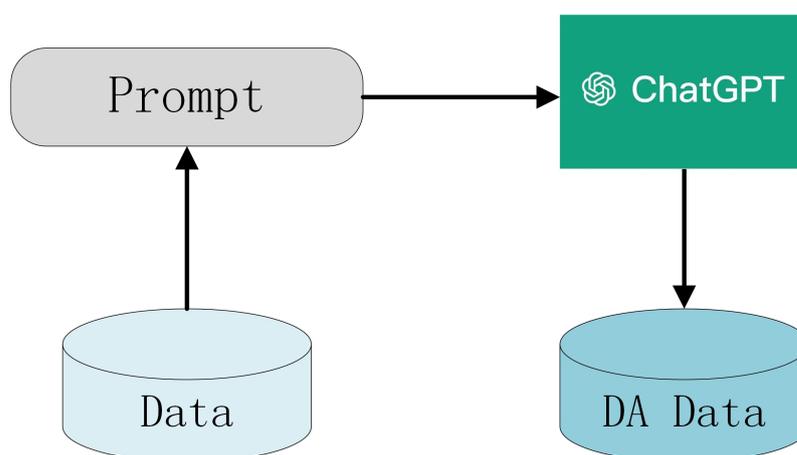
**Table 3.** The original data quantity in the dataset and the number of data instances generated by ChatGPT.

Data	Train	Test
SemEval 2010 Task 8	8000	2717
ChatGPT-Generated	5120	-

If we directly provide the sentence requiring augmentation to ChatGPT and request it to perform data augmentation, several issues may arise:

1. The augmented data may lose special symbols indicating the head and tail entity's beginning and ending positions, such as "<e1>" and "</e1>". This results in the need for entity alignment of the augmented data and then placing special symbols at the beginning and end of the entity, increasing the workload and potentially introducing new noise.
2. The generated augmented data might not adhere to the expected augmentation methods, such as translation, recombination, or entity replacement.
3. The augmented data may suffer from poor quality, and the quantity of generated data may not meet the requirements of this paper.

To address these issues, before formally using ChatGPT for data augmentation, it is necessary to pre-pend a prompt for the sentences to be augmented. For example: "For the sentences: [sentences], please use sentence recombination to generate four augmented data instances in the provided data format". Subsequently, the sentences with the added prompts are provided to ChatGPT to complete the data augmentation task, and the augmented data does not require further modification. The overall process of using ChatGPT for data augmentation is shown in Figure 2.



**Figure 2.** Data augmentation process.

### 3.6. Loss Function

This paper produces three loss functions, where the primary task involves predicting relationship categories, resulting in two loss functions, while the auxiliary task involves predicting relative entity positions, resulting in one loss function. Previous methods for concurrently learning multiple tasks employ a simple weighted sum of losses, where the weights of these losses may be uniform or require manual adjustment:

$$L_{total} = \sum_i W_i L_i \quad (6)$$

Ref. [17] indicates that the performance of multi-task learning highly depends on assigning weights to the loss for each task. A principled approach is proposed that weights multiple loss functions by considering the homoscedastic uncertainty of each task. This paper adopts a similar approach to weight  $Loss_1$ ,  $Loss_2$ , and  $Loss_3$  in the model, resulting in the final loss function:

$$L_{total} = \frac{1}{\sigma_1^2} Loss_1 + \frac{1}{\sigma_2^2} Loss_2 + \frac{1}{\sigma_3^2} Loss_3 + \log \sigma_1 + \log \sigma_2 + \log \sigma_3 \quad (7)$$

where  $\sigma$  is a learnable noise scalar.

## 4. Experiments and Analysis

We first describe the dataset, parameter settings, and evaluation criteria used for the experiments and then explore the impact of each part of the model on model performance. Next, we created three templates to verify the effect of different templates on model performance. We also verified whether the model could misinterpret the relative positions of the entities, which could lead to final prediction errors. Finally, we compare our proposed model with existing models.

### 4.1. Dataset

In the experiments, to evaluate the model, this paper employed a publicly available dataset, SemEval 2010 Task 8. The dataset comprises a training set and a test set, totaling 10,717 samples, with 8000 samples in the training set and 2717 samples in the test set. The dataset includes 9 relationship types and a unique “other” class. These 9 relationship types can be further divided into 18 based on the relative positions of the head and tail entities. For example, “Entity-Origin” can be split into “Entity-Origin (e1, e2)” and “Entity-Origin (e2, e1)”, which are distinct relationship types. Therefore, there are ultimately 19 relationship types in the dataset. The relationships contained in the dataset and the number of each relationship are shown in Table 4.

**Table 4.** The pre-defined number of relationships in the SemEval 2010 Task 8 dataset.

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

### 4.2. Parameter Setting and Evaluation Metrics

This paper evaluates the model using the official scoring script from SemEval 2010 Task 8, with the F1 score as the evaluation metric. We conducted the experiments using the PyTorch [18] deep learning framework. The hardware configuration is an NVIDIA RTX A5000 GPU with 24 GB of memory, an AMD EPYC 7371 CPU. A Warm-Up strategy [19] was employed, which means the model starts training with a minimal learning rate to ensure better convergence. As training progresses, the learning rate gradually increases until it reaches the initial learning rate setting. Subsequently, the learning rate slowly decreases. We set the Warm-Up step to 3200 and initiated the learning rate at  $1 \times 10^{-5}$ . A dropout layer was added to the model to prevent overfitting with a dropout rate of 0.1. We have provided details of the other main parameters of the model in Table 5.

**Table 5.** Parameter setting.

Parameter Name	Value
Number of GCN Layers	2
epoch	12
batch_size	8
Top-K prediction set parameter K	6
hidden_dim	1024
optimizer	AdamW
seq_length	200

#### 4.3. Ablation Experiment

In order to evaluate the performance of the PTKRE models, this section includes ablation experiments on the model. Based on the different pre-trained language models used, these ablation experiments are divided into two control groups: BERT-large and RoBERTa-large. In each ablation experiment group, there are three comparative models:

1. PTKRE-Att model: Change the multi-head attention layer of the PTKRE model that fuses prompt information and Top-K prediction set to average pooling operation. Moreover, we removed the entity location prediction layer.
2. PTKRE-Pos model: We removed only the entity location prediction layer in the PTKRE model.
3. PTKRE-ChatGPT model: Maintain the basic structure of the PTKRE model and remove only the augmented data generated by ChatGPT in the training set.

The results of two sets of ablation experiments using BERT and RoBERTa are shown in Tables 6 and 7. By observing the experimental results of the two control groups, the following conclusions can be drawn:

1. PTKRE vs. PTKRE-Att: After the PTKRE-Att model changes the fusion method of prompt information and Top-K prediction set to average pooling, the model's F1 score decreases in both sets of experiments. It shows that the multi-head attention can more effectively fuse the "<mask>" token in the prompt information and the relationship representation in the Top-K prediction set.
2. PTKRE vs. PTKRE-Pos: After removing the entity location prediction layer from the PTKRE model, the F1 score of the model decreased by 0.14 in both sets of experiments. It shows that the entity position prediction task can effectively assist the model in determining the relative positions between entities in a sentence, thereby enhancing its performance in predicting the relationship categories between entities.
3. PTKRE vs. PTKRE-ChatGPT: After removing the augmented data generated by ChatGPT from the training set, the F1 score of the model decreased noticeably in both control experiments, with reductions of 0.19 and 0.2, respectively. These results indicate that data augmentation significantly impacts the performance of supervised relationship extraction models. The reason for this is that supervised model performance relies heavily on the goodness of the training set, which has high quality and a large amount of data that helps to improve the generalization of the model.

**Table 6.** Results of ablation experiments using BERT.

Model	F1-Score
PTKRE-Att	90.19
PTKRE-Pos	90.31
PTKRE-ChatGPT	90.26
PTKRE	90.45

**Table 7.** Results of ablation experiments using RoBERTa.

Model	F1-Score
PTKRE-Att	91.10
PTKRE-Pos	91.24
PTKRE-ChatGPT	91.18
PTKRE	91.38

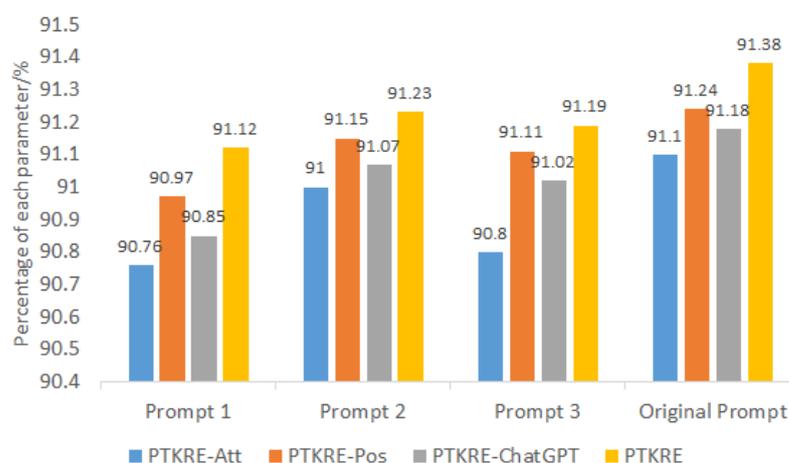
#### 4.4. Impact Assessment of Different Prompt Templates

In this paper, we add prompt information before each sentence, and to evaluate the influence of different prompt templates on the performance of the PTKRE model, we design three new prompt templates as follows:

- Original Prompt: [ent1] and [ent2] are related in the sentence through <mask> <mask> sentence:
- Prompt 1: In this sentence, the relation is <mask> <mask>:
- Prompt 2: [ent1] and [ent2] are related in the sentence through <mask> <mask>
- Prompt 3: In this sentence, the relationship between [Ent1] and [Ent2] is <mask> <mask> sentence:

In the experiments, we used RoBERTa as the pre-trained language model and explored the impact of different prompt templates on the performance of the PTKRE model. The relevant experimental results are presented in Figure 3:

1. Template 1 removed entity-related information from the original prompt, retaining only the "<mask>" token and the sentence starting token ".". The experimental results show that removing entity-related information impairs the model's performance, resulting in an overall F1 score decrease of 0.26 to 0.34 compared to the original template.
2. Template 2, based on the original template, removed sentence starting information "sentence:", which indicated the starting position of the sentence where RoBERTa extracts entity relationships from the data. Compared to the original template, the experimental results show that Template 2's overall F1 score decreased by 0.09 to 0.15, indicating relatively more minor damage to the model than removing entity-related information.
3. Template 3 retained all the information but used a different phrasing while conveying the same meaning as the original template. Template 3's overall F1 score decreased by 0.13 to 0.3 compared to the original template.



**Figure 3.** Impact of different prompt on F1.

These experimental results illustrate that different prompt templates significantly impact the model's final performance. This study confirms that adding entity-related information and sentence-starting position information to the prompt template enhances the model's performance. Furthermore, different phrasings also influence the model's performance.

#### 4.5. Entity Location Prediction Experiment

Due to the specificity of the predefined relationship classes in the SemEval 2010 Task 8 dataset, this section verifies experimentally whether the model confuses the relative positions of two entities. We conducted experiments using the PTKRE-Pos model, which we obtained by removing only the entity position prediction layer from the PTKRE model.

We have displayed two experimental results of the model predictions in Table 8. Observing the prediction results of the PTKRE-Pos model on the test set, we can see that this model confuses the relative positions of the two entities, which results in incorrect predictions. Then, we employed the PTKRE model to predict the test set once more, and we compared the prediction results with those of the PTKRE-Pos model. The results show that with the addition of the entity location prediction layer, the model can accurately

determine the relative locations of entities, thus obtaining correct prediction results. These experimental results show that the entity location prediction layer can effectively assist the model in recognizing the entity location, thus improving the model's performance.

**Table 8.** Comparison of correct results and model predictions.

Sentence:	His trademark steam-engine puffing is revealed as a <e1> sound </e1> made by a <e2> viper </e2> spitting venom at his prey before swallowing her whole.
Truth label:	Cause–Effect (e2, e1)
PTKRE-Pos Predictive labels:	Cause–Effect (e1, e2)
PTKRE Predictive labels:	Cause–Effect (e2, e1)
Sentence:	A “green bean” which is actually a <e1> fruit </e1> with <e2> seeds </e2> inside.
Truth label:	Component–Whole (e1, e2)
PTKRE-Pos Predictive labels:	Component–Whole (e2, e1)
PTKRE Predictive labels:	Component–Whole (e1, e2)

#### 4.6. Comparison of Different Methods

In order to validate the effectiveness of the PTKRE model on the relational extraction dataset, we conducted this part of the experiment on the SemEval 2010 Task 8 dataset. We compared it with other relational extraction models.

1. The R-BERT [11] model adds different special labels for the head and tail entities, enriching the pre-trained BERT model by using the entity information for the relationship classification task.
2. The A-GCN [8] model utilizes dependency information for relationship classification. The attention mechanism is applied to dependency connections by assigning weights for both connections and types to distinguish the importance of dependency information better.
3. The Skeleton-Aware BERT [20] model proposes an indicator-aware relation extraction method in order to be able to utilize both syntactic indicators and sentence context. First, this model extracts the syntactic indicators under the guidance of syntactic knowledge. Then, a neural network is constructed to combine the syntactic indicators and the whole sentence to represent the relation better.
4. The KLG [16] model utilizes Top-K prediction sets to improve performance on the relational extraction task. First, a pre-trained language model is fine-tuned on the downstream dataset, and this PLM automatically generates Top-K prediction sets for each sample, where a dynamic K selection mechanism generates K. Then, a labeled graph neural network is constructed.
5. The PTR [13] method is a prompt-based learning approach that proposes to encode the a priori knowledge of a classification task into rules, then design sub-prompts based on the rules and apply a masked training task of a language model to predict the classification.
6. The RIFRE [21] model proposes an iterative fusion method for representations based on heterogeneous graph neural networks. The method takes relations and words as nodes on the graph. It iteratively fuses the two types of semantic nodes through a message-passing mechanism to get a representation of the more suitable node for the relation extraction task. When the node representation is updated, the model does the relation extraction task.

As seen from the experimental results in Table 9, the F1 score of the PTKRE model is improved compared to the comparison models. R-BERT is a more classical relational extraction model after the emergence of the pre-trained language model BERT, and the PTKRE model improves the F1 score by 2.13 compared to the R-BERT model, which illustrates the effectiveness of the PTKRE model in the relational extraction task.

**Table 9.** Results of model comparison experiments on the SemEval 2010 Task 8 dataset.

Model	F1-Score
R-BERT	89.25
A-GCN	89.85
PTR	89.9
Skeleton-Aware BERT	90.36
KLG	90.5
RIFRE	91.3
PTKRE	91.38

## 5. Conclusions

This paper introduces a relation extraction model incorporating prompt information and the Top-K prediction set. Additionally, ChatGPT is employed to augment the training dataset, thereby improving the performance of the relation extraction model. The incorporation of prompt information aims to reduce the disparities between the pre-trained language model RoBERTa during its pre-training and fine-tuning stages, allowing for a more comprehensive utilization of the semantic information provided by the RoBERTa model. Furthermore, we observed that the predefined relationship categories in the SemEval-2010 Task 8 dataset are contingent on the relative positions of the head and tail entities. This dependency manifests as a challenge during experiments, where models may confuse entity positions, leading to erroneous results. To address this issue, we propose an entity position prediction task that assists the model in accurately identifying the relative positions of entities within sentences. As a result, the approach presented in this paper achieves an F1-score of 91.38 on the SemEval 2010 Task 8 dataset. In the future, we will focus on how to auto-generated prompts since, in our experiments, we found that different prompts impact the model's performance. Data augmentation is significant for supervised relational extraction models, and we will continue to explore how to utilize ChatGPT to accomplish data augmentation tasks better.

**Author Contributions:** Conceptualization, P.F. and D.O.; methodology, P.F. and H.W.; software, Y.W. and Z.Y.; validation, P.F., D.O. and H.W.; writing—original draft preparation, H.W.; writing—review and editing, P.F. and H.W.; visualization, P.F. and H.W.; funding acquisition, P.F. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the Science and Technology Development Plan Project of Jilin Provincial Science and Technology Department (Key Technology Research on Risk Prediction and Assessment of Old Chronic Diseases Based on Medical Knowledge Graph (2023JB405L07)).

**Data Availability Statement:** Publicly available datasets were used in this study. This data can be found here: <http://www.kozareva.com/downloads.html> (accessed on 2 November 2023).

**Acknowledgments:** We would like to express our deepest gratitude to all those who have contributed to the completion of this research and the writing of this paper.

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

## Abbreviations

The following abbreviations are used in this manuscript:

CNN	Convolutional Neural Networks
GCN	Graph Convolutional Networks
PLM	Pre-trained Language Model
MLP	Multilayer Perceptron
BERT	Bidirectional Encoder Representations from Transformers
RoBERTa	Robustly Optimized BERT Pretraining Approach
NLP	Natural Language Processing

## References

1. Kenton, J.D.M.W.C.; Toutanova, L.K. Bert: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the NAACL-HLT, Minneapolis, MN, USA, 2–7 June 2019; Volume 1, p. 2.
2. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv* **2019**, arXiv:1907.11692.
3. Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I. *Improving Language Understanding by Generative Pre-Training*; OpenAI: San Francisco, CA, USA, 2018.
4. Liu, P.; Yuan, W.; Fu, J.; Jiang, Z.; Hayashi, H.; Neubig, G. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.* **2023**, *55*, 1–35. [[CrossRef](#)]
5. 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 30: Annual Conference on Neural Information Processing Systems 2017, Long Beach, CA, USA, 4–9 December 2017.
6. Zeng, D.; Liu, K.; Lai, S.; Zhou, G.; Zhao, J. Relation classification via convolutional deep neural network. In Proceedings of the COLING 2014 25th International Conference on Computational Linguistics: Technical Papers, Dublin, Ireland, 23–29 August 2014; pp. 2335–2344.
7. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907.
8. Guo, Z.; Zhang, Y.; Lu, W. Attention guided graph convolutional networks for relation extraction. *arXiv* **2019**, arXiv:1906.07510.
9. Mandya, A.; Bollegala, D.; Coenen, F. Graph Convolution over Multiple Dependency Sub-graphs for Relation Extraction. In Proceedings of the COLING, International Committee on Computational Linguistics, Barcelona, Spain, 8–13 December 2020; pp. 6424–6435.
10. Xiong, S.; Li, B.; Zhu, S. DCGNN: A single-stage 3D object detection network based on density clustering and graph neural network. *Complex Intell. Syst.* **2022**, *9*, 3399–3408. [[CrossRef](#)]
11. Wu, S.; He, Y. Enriching pre-trained language model with entity information for relation classification. In Proceedings of the 28th ACM international Conference on Information and Knowledge Management, Beijing, China, 3–7 November 2019; pp. 2361–2364.
12. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language models are few-shot learners. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 1877–1901.
13. Han, X.; Zhao, W.; Ding, N.; Liu, Z.; Sun, M. Ptr: Prompt tuning with rules for text classification. *AI Open* **2022**, *3*, 182–192. [[CrossRef](#)]
14. Feng, P.; Zhang, X.; Zhao, J.; Wang, Y.; Huang, B. Relation Extraction Based on Prompt Information and Feature Reuse. *Data Intell.* **2023**, *5*, 824–840. [[CrossRef](#)]
15. Yu, J.; Zhu, T.; Chen, W.; Zhang, W.; Zhang, M. Improving relation extraction with relational paraphrase sentences. In Proceedings of the 28th International Conference on Computational Linguistics, Barcelona, Spain, 8–13 December 2020; pp. 1687–1698.
16. Li, B.; Ye, W.; Zhang, J.; Zhang, S. Reviewing labels: Label graph network with top-k prediction set for relation extraction. In Proceedings of the AAAI Conference on Artificial Intelligence, Washington, DC, USA, 7–14 February 2023; Volume 37; pp. 13051–13058.
17. Kendall, A.; Gal, Y.; Cipolla, R. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7482–7491.
18. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. Pytorch: An imperative style, high-performance deep learning library. In Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, BC, Canada, 8–14 December 2019; Volume 32.
19. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
20. Tao, Q.; Luo, X.; Wang, H.; Xu, R. Enhancing relation extraction using syntactic indicators and sentential contexts. In Proceedings of the 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI), Portland, OR, USA, 4–6 November 2019; pp. 1574–1580.
21. Zhao, K.; Xu, H.; Cheng, Y.; Li, X.; Gao, K. Representation iterative fusion based on heterogeneous graph neural network for joint entity and relation extraction. *Knowl.-Based Syst.* **2021**, *219*, 106888. [[CrossRef](#)]

**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.