Generating Diverse Training Samples for Relation Extraction with Large Language Models

Zexuan Li, Hongliang Dai*, Piji Li
[1] College of Artificial Intelligence,
Nanjing University of Aeronautics and Astronautics, Nanjing, China
[2] MIIT Key Laboratory of Pattern Analysis and Machine Intelligence, Nanjing, China
[3] The Key Laboratory of Brain-Machine Intelligence Technology, Ministry of Education, Nanjing, China.
{zexuanli, hongldai, pjli}@nuaa.edu.cn

Abstract

Using Large Language Models (LLMs) to generate training data can potentially be a preferable way to improve zero or few-shot NLP tasks. However, many problems remain to be investigated for this direction. For the task of Relation Extraction (RE), we find that samples generated by directly prompting LLMs may easily have high structural similarities with each other. They tend to use a limited variety of phrasing while expressing the relation between a pair of entities. Therefore, in this paper, we study how to effectively improve the diversity of the training samples generated with LLMs for RE, while also maintaining their correctness. We first try to make the LLMs produce dissimilar samples by directly giving instructions in In-Context Learning (ICL) prompts. Then, we propose an approach to fine-tune LLMs for diversity training sample generation through Direct Preference Optimization (DPO). Our experiments on commonly used RE datasets show that both attempts can improve the quality of the generated training data. We also find that comparing with directly performing RE with an LLM, training a non-LLM RE model with its generated samples may lead to better performance.

1 Introduction

Relation Extraction (RE) aims to identify and classify specific relation categories between pairs of entities from text. It is an important task in information extraction and has been deeply used in knowledge graph construction (Zhong et al., 2024), question and answer systems (Srihari and Li, 2000) and so on. Existing models (Chen et al., 2022b; Zhou and Chen, 2022; Paolini et al., 2021) applied to RE have achieved good results on many benchmarks. However, since there are various types of entity relations in different domains, data scarcity is a common problem while developing RE models in practice. Existing studies typically address this problem with techniques such as prototypical networks (Liu et al., 2022), meta-learning (Qu et al., 2020) and prompt-tuning (Chen et al., 2022b) under zero or few-shot settings.

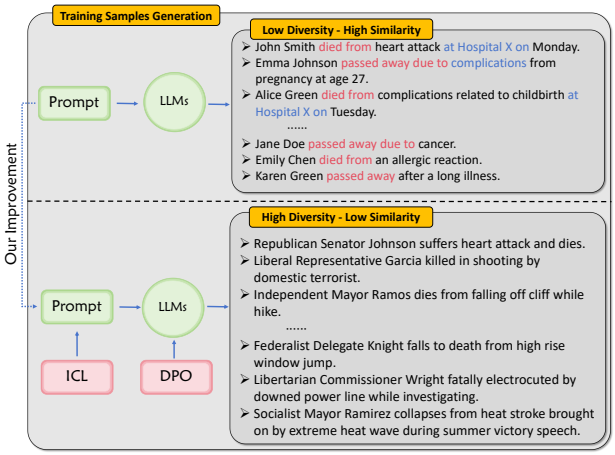

Figure 1: Training samples generated by LLMs for RE before and after adopting our approach.

Recently, the powerful generative capabilities of Large Language Models (LLMs) (OpenAI, 2022, 2023; Brown et al., 2020) have made it possible to alleviate the data scarcity problem with a new way: using LLMs to generate extra training data for the task. In this paper, we study the application of this approach to relation extraction. We find that directly prompting LLMs to obtain RE training samples may easily lead to low diversity results. An example is shown in the upper part of Figure 1, the LLM repeatedly uses a limited number of key verbs or phrases to indicate the relation between the head and tail entities. The structures of the sentences are also almost the same.

We therefore focus on how to improve the diversity of LLM generated training instances for RE, while maintaining the correctness. To this end, several approaches are proposed and tried. First, we employ the In-Context Learning (ICL) (Wan

*Corresponding author.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 713–726

et al., 2023; Dong et al., 2023; Brown et al., 2020) technique. The most straightforward way to improve diversity through ICL is to directly give the model an extra instruction in the prompt, asking it to generate dissimilar samples. In addition, we also propose a one by one generation procedure, where each time, the model is instructed to generate one more training sample that is different from the given demonstrations.

Conducting ICL cannot change the inherent behavior of LLMs. Thus, we also propose an approach to fine-tune LLMs for diversity RE sample generation. We adopt Direct Preference Optimization (DPO) for fine-tuning, which has been verified to perform well on many tasks such as summary generation and single-round conversations (Rafailov et al., 2023). RE samples that are similar with existing ones and incorrect samples are automatically generated to serve as dispreferred answers to the DPO algorithm, thus training the LLM to consider both diversity and correctness. The lower part of Figure 1 is an example of the training instances obtained through our approach. It can be seen that these samples generated by the LLMs are more diverse in terms of the verbs or phrases that indicate relations, and the overall architectures of the sentences.

Finally, we conduct extensive experiments on commonly used RE datasets including three versions of TACRED and SemEval to evaluate the effectiveness of our method. We also show that comparing with directly performing RE with an LLM, training a non-LLM RE model with its generated samples can potentially lead to better performance.

Our main contributions as summarized follows:

- We investigate ICL-based methods for diverse RE training sample generation with two different procedures: *one by one* and *all at once*.
- We propose an approach to fine-tuning LLMs with DPO that aims for generating diverse and correct RE training samples. The approach uses automatically constructed dispreferred answers, therefore reduces the requirement of human annotation.
- We provide comprehensive experimental results to analyze the performance of the proposed methods.

Our code is available at https://github.com/Lzx-ZBC.

2 Related Work

2.1 Relation Extraction with LLMs

Relation Extraction (RE) aims to extract the relationship between head and tail entities based on their relevant context. The task can be approached by using traditional neural network models such as CNN and RNN (Zeng et al., 2015; Zhang et al., 2017; Zhou et al., 2016). Since the proposal of Pre-Trained Language Models (PLMs) like BERT (Devlin et al., 2019), PLMs-based RE models has become the main solution due to their preferable performance. Alternatively, graph neural networks can also be employed for conducting RE (Guo et al., 2020, 2019; Zhang et al., 2018).

Recently, there is an increasing interest in using LLMs to perform RE directly. Xu et al. (2023) put a list of all relation categories and concrete samples in the prompt to make LLMs understand the exact process of RE. Their experimental results show that LLMs is capable of producing high quality RE predictions. Zhang et al. (2023) propose a framework called QA4RE that coordinates RE with question and answer (QA). Li et al. (2023) proposes a new prompting method, SUMASK, which converts the input into a valid QA format using LLMs by decomposing RE into text summarization and QA.

2.2 Data Generation with LLMs

Data generation is gradually becoming a new focus topic. There were already some studies on this topic before instruction tuned LLMs become popular (Meng et al., 2022; Ye et al., 2022; Gao et al., 2023). For example, Meng et al. (2022) generates training data for NLU tasks such as sentiment classification through prompting PLMs. With instruction tuned LLMs, data generation becomes more convenient. Chia et al. (2022) proposes a framework to synthesize unseen relation types by prompting language models to generate structured text. Xu et al. (2023) use LLMs to generate data to assist the models themselves on RE. They accomplish this by describing the data content and samples in detail in the prompt so that LLMs can generate reasonable data. Recently, as instruction tuned LLMs have entered the research field, generating instruction tuned datasets requires careful writing of instructions and input-output pairs, which are usually written by humans, smaller in size and less in diversity. To overcome this problem, self-instruct (Wang et al., 2023) proposes a method for generating instruction tuned datasets by

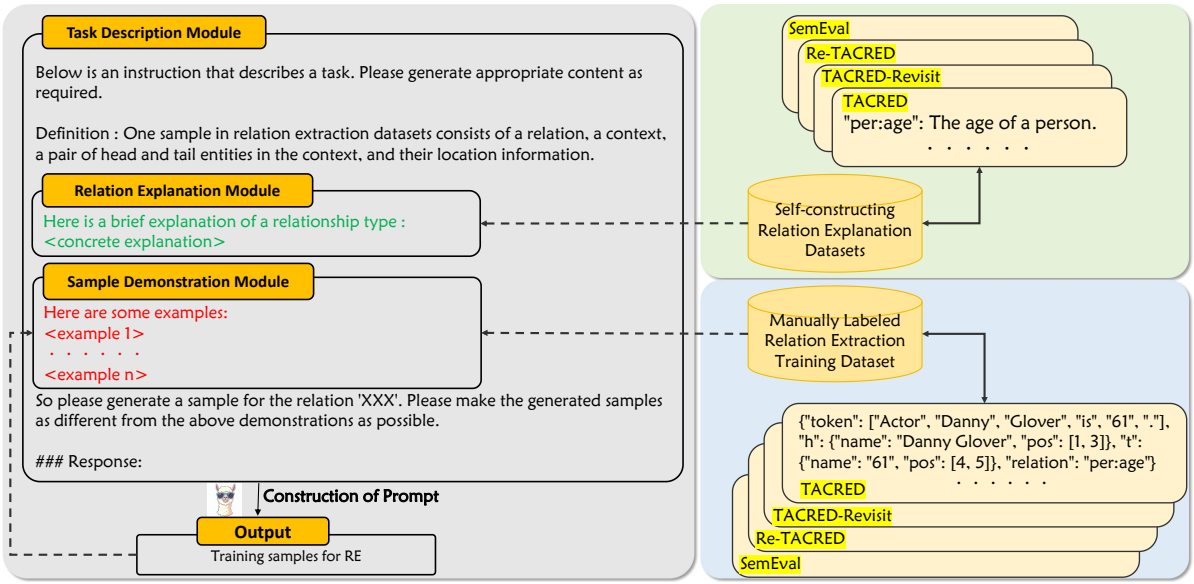

Figure 2: Construction of Prompt, which consists of three modules: Task Description Module, Relation Explanation Module and Sample Demonstration Module.

prompting available LLMs. The results show that the data generated by LLMs can improve their classification ability on RE. However, they do not put a specific focus on how the diversity and correctness of the generated training samples can be improved.

3 Methodology

In this section, we first introduce how we generate training samples for RE by directly performing in-context learning with LLMs. Then, we describe our approch to fine-tune LLMs for diverse and correct sample generation with DPO.

3.1 Sample Generation with ICL

To apply in-context learning (ICL), we provide sample demonstrations for LLMs in the prompt to stimulate their understanding of the training instances for the relation extraction task. With the demonstrations, the straightforward way to prompt LLMs is to instruct them to directly generate the training samples *all at once*. Here, in addition, we adopt an alternative way. We generate training samples *one by one* instead of *all at once*. Doing so enables the training samples generated by the LLMs to be added back to the prompt, allowing us to instruct the model not to generate new samples similar with the already existing ones. Next, we introduce this method in detail. The method of generating *all at once* will be explained in the experimental Results and Analysis section. The prompt for LLMs constructed by us are shown in Figure 2. It includes three modules: task description, relation explanation and sample demonstration. Their specific contents are as follows:

Task Description Module. The Task Description Module covers the beginning and the end of the full prompt. At the beginning of the prompt, we tell LLMs that it will accomplish the task of generating training samples for relation extraction, and detail that a sentence-level training sample for RE includes the key information of text, head and tail entities with their locations, and the relation category. At the end of the prompt, we order LLMs to generate new training samples for the relation category described in the Relation Explanation Module, and require that the generated training samples be as different as possible from the provided demonstrations in the Sample Demonstration Module.

Relation Explanation Module. We construct a corresponding explanation for each relation category in the relation extraction dataset, and then put them into the Relation Explanation Module to allow the LLMs to further understand the relation type. The details of the explanations for all relation categories are provided in Appendix A.2.

Sample Demonstration Module. For a specific relation category, we randomly select a manually labeled sample from the existing relation extraction dataset and put it into the Sample Demonstration Module for LLMs' reference, so that LLMs can further understand the structure and content of the

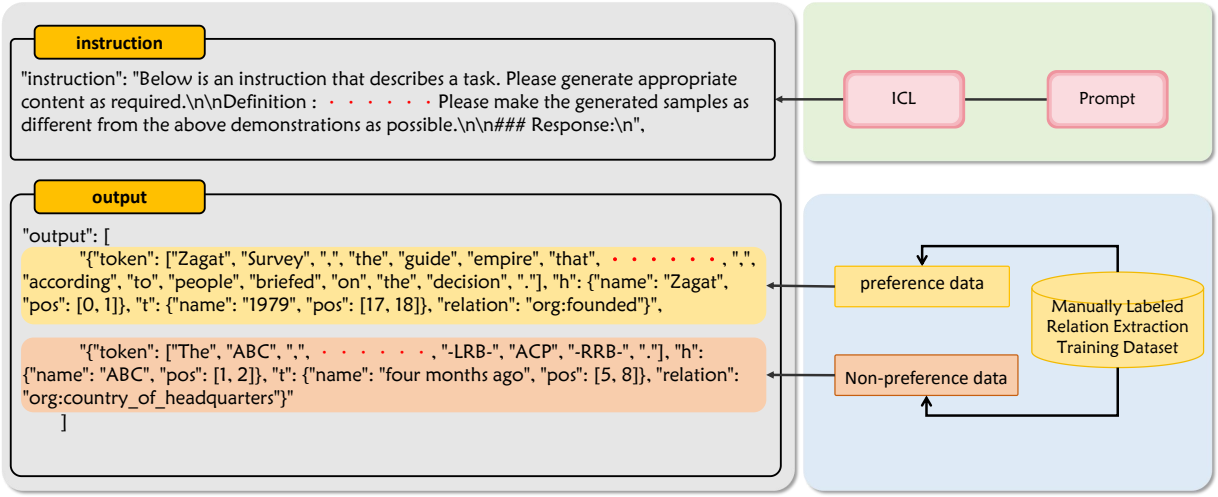

Figure 3: Construction of the DPO Fine-Tuning Training Dataset.

relation extraction training samples that need to be generated. Once LLMs have completed one generation, the generated training instance is moved back to the Sample Demonstration Module to form a new Prompt. Under cyclic generation, the number of demonstration samples in this module will increase one by one as the number of generated samples increases.

3.2 Diversity Fine-tuning with DPO

Direct Preference Optimization (DPO) is an automated fine-tuning method that optimizes model parameters by maximizing the rewards of a pre-trained model on a specific task. Compared to traditional fine-tuning methods, DPO bypasses the step of modeling the reward function and instead improves performance by optimizing the model directly on the preference data. Given a human preference dataset $\mathcal{D} = \{(x_i, y_{i,1}, y_{i,2})\}$, where $y_{i,1}$ is preference data, $y_{i,2}$ is non-preference data, the following objective can be optimized:

$$\max_{\pi} \Sigma_i \log \sigma \left(\frac{1}{\beta} \log \frac{\pi(y_{i,1}|x_i)}{\pi(y_{i,2}|x_i)} \right), \qquad (1)$$

where $\pi(y|x)$ is the optimization strategy. According to this equation, the key to DPO fine-tuning is to construct suitable preference and non-preference data for LLMs to make comparisons.

Construction of the DPO Fine-Tuning Data. Each training instance for DPO consists of an input prompt ("instruction"), as well as two responses ("output"): a preferred response and an non-preferred response. As shown in Figure 3, we use the previously described ICL prompt as "instruction". Both the preference and non-preference data will be placed in the "Output". The preference data is a random sample of the target relation category directly extracted from the manually labeled dataset. The non-preference data is constructed in the following three ways: (1) Use a manually labeled training sample that belongs to another relation category, and modify its label to the target relation category. This constructs an sample that is incorrectly annotated as the target relation. (2) Choose a demonstration sample in the "Instruction" item, then replace the head and tail entities, and add or delete a few words in the context. This creates an instance that is similar to one of the demonstration samples. (3) Use a sample that is exactly the same as one of the demonstration samples in the "Instruction" item.

Here, note that since we focus on the data scarcity scenario, there won't be enough manually annotated data of the target relation types for DPO fine-tuning. Thus, the source data for DPO should come from other already existing datasets that probably use different relation types or are of different domains. In our experiments, to mimic such a scenario, we split the datasets by relation categories and use different parts for DPO and sample generation, respectively. This will be detailed in the Experimental Settings section.

Imitating One by One Generation. As shown in Figure 4, in order for the DPO fine-tuning to reflect the one by one generation process, for each relation category, we let the number of demonstra-

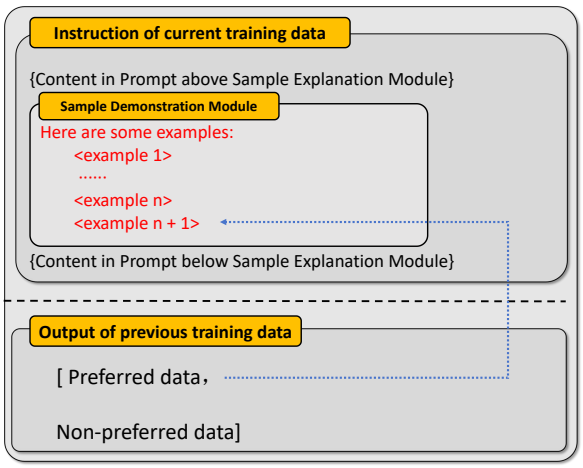

Figure 4: Imitating one by one generation during Direct Preference Optimization.

tion samples in the "Instruction" item of the DPO fine-tuning training dataset increment, specifically by placing the preferred manually labeled training samples from the "Output" item of the previous training data into the Sample Demonstration Module of the current training data while forming the "Instruction" item of new training data.

4 Experimental Settings

4.1 Datasets

We conduct experiments on SemEval and three versions of TACRED: SemEval 2010 Task 8 (SemEval) (Hendrickx et al., 2010), TACRED (Zhang et al., 2017), TACRED-Revisit (Alt et al., 2020), Re-TACRED (Stoica et al., 2021). Statistical details are given in Table 1 and Appendix A.1.

Dataset	Train	Val	Test	Relation
SemEval	6,507	1,493	2,717	19
TACRED	68,124	22,631	15,509	42
TACRED-Revisit	68,124	22,631	15,509	42
Re-TACRED	58,465	19,584	13,418	40

Table 1: Statistics of the RE datasets. Including the numbers of instances in different splits and the numbers of relations.

4.2 Evaluation

Considering that it is difficult to directly assess the quality of the generated training samples, we put the generated training samples into the KnowPrompt (Chen et al., 2022b) and RetrievalRE (Chen et al., 2022a) for training. KnowPrompt achieves satisfying performance by Knowledge Injection and Synergistic Optimization. RetrievalRE, on the other hand, improves the generalization ability of the model when dealing with difficult patterns by combining retrieval enhancement and prompt tuning. Their performance on the test dataset reflects the quality of the generated training samples. The better the performance of them, the higher the quality of the generated training samples.

Finally, we follow existing RE studies and adopt Micro F1 as the evaluation metric.

4.3 Implementation Details

For KnowPrompt and RetrievalRE, we follow (Chen et al., 2022b,a) and use RoBERTA_LARGE (Liu et al., 2019) in all the experiments for a fair comparison. For LLMs, considering the cost and fine-tuning requirements, we used LLaMA2-7b-Chat (Touvron et al., 2023) in our experiments. We set temperature = 0.4, top_p = 0.9, top_k = 20, repetition_penalty = 1.15. Meanwhile, we use LoRA to accomplish DPO fine-tuning of LLMs. We set truncation length = 1024, learning rate = 5e-5, batch size = 4, epoch = 20.

In order to prevent the risk of "cheating" on LLMs generation caused by DPO fine-tuning, we separated the relation categories used for DPO fine-tuning from those to be generated. Specifically, for each dataset, we divide the relation categories in half, thereby also partition the dataset into two parts. Then one part is used for DPO fine-tuning, and the other part is used for sample generation. By swapping the two parts, we are able to complete the sample generation of all relation categories.

5 Results and Analysis

5.1 Main Results

We compare with directly conducting RE using the LLM through ICL, and directly training KnowPrompt and RetrievalRE using only manually labeled data. The results are in Table 2. For DirectRE, 2 samples are used as demonstrations. For our methods, 64 samples are generated for training. Note that in practice, more generated samples can be used if the performance can be further improved. An analysis over the number of generated samples is provided in Section 5.4.

Comparing to Manually Labeled Training Samples. In Table 2, Ours (pure) only uses the LLM-generated samples for training, while KnowPrompt and RetrievalRE only uses the manually labeled samples. The performance of Ours (pure) is comparable to that of KnowPrompt and RetrievalRE on

Method	TACRED			TACRED-Revisit			Re-TACRED			SemEval		
	K=8	K=16	K=32	K=8	K=16	K=32	K=8	K=16	K=32	K=8	K=16	K=32
Direct-RE		21.17			21.82			33.30			25.55	
Data Generation	21.67	24.71	26.19	23.14	27.89	29.80	28.52	32.21	32.04	-	-	-
KnowPrompt	22.07	30.00	**36.33**	26.31	30.76	34.61	44.77	56.51	62.34	61.39	74.08	**81.06**
Ours (pure)	22.48	27.99	30.41	23.04	29.25	31.12	34.77	40.33	50.72	42.02	45.05	47.70
Ours (mix-OBO)	33.48	34.93	36.25	**34.33**	34.39	35.27	**57.41**	**61.09**	**64.58**	67.92	75.36	80.40
Ours (mix-AAO)	**34.30**	**35.56**	35.66	34.15	**34.77**	**35.65**	56.01	59.97	63.55	**70.32**	**76.36**	80.46
Ours (constant)	17.39	21.71	-	18.39	22.22	-	36.01	39.53	-	22.32	30.53	-
Data Generation	24.97	30.54	25.99	27.67	28.46	29.48	31.21	35.03	35.94	-	-	-
RetrievalRE	32.04	35.16	37.07	28.95	32.34	**37.26**	33.73	55.79	61.80	72.53	**81.42**	83.93
Ours (pure)	25.20	29.63	30.29	28.55	31.73	31.51	27.92	39.21	50.05	45.13	47.20	49.22
Ours (mix-OBO)	33.75	35.83	**37.46**	33.81	34.15	36.90	**58.60**	**62.61**	**65.15**	**74.20**	80.84	**84.01**
Ours (mix-AAO)	**34.91**	**36.57**	37.23	**35.06**	**35.64**	36.77	55.43	61.15	64.85	73.36	78.90	81.22
Ours (constant)	22.48	25.31	-	23.99	26.38	-	33.28	41.48	-	36.65	39.47	-

Table 2: Micro F1 (%) of few-shot performance. **Direct-RE** means using LLMs directly for RE. **KnowPrompt** and **RetrievalRE** means the performance of manually labeled training samples. **Ours (pure)** means the performance of using LLM-generated training samples only. **Ours (mix-OBO)** and **Ours (mix-AAO)** mean the performance of combining the use of manually labeled training samples and training samples generated by LLM based on the OBO and AAO. **Ours (constant)** means the performance of LLMs-generated training samples with a fixed number of demonstration samples in Prompt. **Data Generation** means the performance of training samples generated by (Xu et al., 2023).

TACRED and TACRED-Revisit, but is much worse on Re-TACRED and SemEval. Therefore, the qualities of the generated samples are good enough to be beneficial for training. But they still cannot be used to for the purpose of fully replacing manually labeled data, even when the number of training samples is as small as 8.

Comparing to Generated Training Sample by Another Way. As demonstrated in Table 2, Ours (Pure) performs the baseline (Data Generation) across most experimental settings on the three variants of TACRED, indicating the superior efficacy of our sample generation methodology.

Comparing to Performing Relation Extraction Directly with LLMs. As shown in Table 2, Ours (pure) performs better than directly using LLM to conduct relation extraction (Direct-RE) on all four relation extraction datasets. This means that when people employ LLMs for RE without any manually labeled data, they can consider training a non-LLM model with LLM-generated samples, instead of directly prompting the LLMs. This would also reduce the cost of repeatedly calling LLMs. Moreover, the self-improvements of LLMs on the RE task can also be an interesting direction.

Comparing to Mixed Training Samples. We mix manually labeled training samples with LLMs-generated training samples and use them for the training of KnowPrompt and RetrievalRE. This corresponds to Ours (mix-OBO, mix-AAO) in Table 2, where K denotes the number of manually labeled training samples. The number of LLMs-generated training samples varies dynamically, and we choose the value in [8, 16, 32] that gives the best performance for KnowPrompt and RetrievalRE. We find that the model's performance is higher than when only pure LLM-generated training samples or only pure manually labeled training samples are used under most of the settings. This shows that the generated samples can be combined with existing human annotated data to help improve the final performance of RE models. However, as the amount of human annotated training data increases, the generated samples become less beneficial.

Comparing to Using a Constant Number of Demonstration Samples. We fixed the number of samples in the Sample Demonstration Module to 4. When the LLMs generate a new training sample, and if the number of samples demonstrated in the Sample Demonstration Module does not reach 4, the newly generated samples are directly added to this Module. Conversely, if the number reaches 4, we let it randomly replace one of the samples in the Sample Demonstration Module. As shown in Table 2, after the training samples generated

Method	TACRED			TACRED-Revisit			Re-TACRED			SemEval		
	K=8	K=16	K=32	K=8	K=16	K=32	K=8	K=16	K=32	K=8	K=16	K=32
Ours (AAO)	**21.86**	26.91	26.39	**23.17**	27.97	28.50	**38.00**	**45.68**	**46.50**	**41.49**	**54.53**	**56.34**
Ours (AAO), w/o DPO	18.97	**27.70**	**36.04**	22.61	**31.08**	**37.08**	20.64	36.68	36.66	37.66	47.49	49.58
Ours (OBO)	**22.48**	**27.99**	**30.41**	**23.04**	**29.25**	**31.12**	**34.77**	40.33	**50.72**	**42.02**	**45.05**	**47.70**
Ours (OBO), w/o DPO	18.35	21.47	23.70	18.39	24.16	27.52	31.82	**41.47**	40.45	24.65	28.42	30.03
Ours (OBO), w/o DPO, w/o DI	21.86	21.75	25.66	21.90	24.11	27.90	33.50	33.55	32.74	20.97	27.01	25.88

Table 3: Micro F1 (%) of different variants of our approach on KnowPrompt. **AAO** means generating all training samples at once. **OBO** means generating training samples one by one. **DI** means the instruction about "diversity". The best results in each column are indicated by **bolding**.

under this approach are used for KnowPrompt or RetrievalRE training, the performance of the models drop substantially. We found by tracking the generation process of LLMs that when a training sample in the Sample Demonstration Module is replaced, the LLMs have a high probability of generating instances similar or even identival to this sample in the subsequent period, thus leading to a poor overall quality of the generated samples.

5.2 Ablation Analysis

In order to verify the effectiveness of DPO, as well as comparing with generating samples *all at once*, we complete the ablation experiments for the following three different variants:

Generate all at once without DPO. We change the key sentence in the Prompt for LLMs to *"So please generate 32 samples for the relation '$\mathcal{R}$'. Please make the generated samples as different from the above demonstrations as possible."*.

Generate one by one without DPO. We change the key sentence in the Prompt for LLMs to *"So please generate a sample for the relation '$\mathcal{R}$'. Please make the generated samples as different from the above demonstrations as possible."*.

Generate one by one without DPO & diversity instruction. We change the key sentence in the Prompt for LLMs to *"So please generate a sample for the relation '$\mathcal{R}$'."*.

As shown in Table 3, in most cases, either in AAO or OBO generation mode, the performance of the generated training samples on the KnowPrompt mostly decrease after removing DPO fine-tuning, which indicates that reasonable DPO fine-tuning can help LLMs generate higher-quality training samples. Furthermore, it can be seen that the samples generated by OBO perform better on the RE models than those generated by AAO In most experimental settings, indicating that our proposed OBO generation mode is effective and can further improve the quality of training samples. Meanwhile, comparing the results of the experiments under the "w/o DPO, AAO" and "w/o DPO, OBO" conditions, it can be seen that the quality of the training samples under the former condition is higher. We think it is because in the "OBO" condition, the number of samples demonstrated in the Sample Demonstration Module in the Prompt increases, which reduces the attention to "diversity instruction", and makes the LLMs show "Imitation Behavior" when generating samples, resulting in higher similarity and lower quality of the final generated training samples. Finally, comparing the results of the experiments in the "w/o DPO, OBO" and "w/o DPO, w/o DI, OBO" conditions, it can be seen that the quality of the generated training samples is higher in the former condition, which suggests that the "diversity instruction" can remind the LLMs to take diversity into account when generating training samples. Also note that in the case of "Re-TACRED, k=16", the training samples generated by LLMs without DPO fine-tuning have poor diversity, but they perform better on Knowprompt model than the training samples generated by LLMs with DPO fine-tuning, which we think is related to the test data of Knowprompt model. The training samples generated by LLMs without DPO fine-tuning may just be close to the test data, so there is a phenomenon of low diversity but good training effect.

5.3 Diversity of Generated Samples

We extract the value of the "token" key from the sentence-level samples generated by LLMs, and then restore the words into sentences to facilitate similarity calculation. We calculate the average cosine similarity between every two pairs of generated training samples through OBO for each relation category, and the results are shown in Figure 5. **The overall results show that the diversity of the**

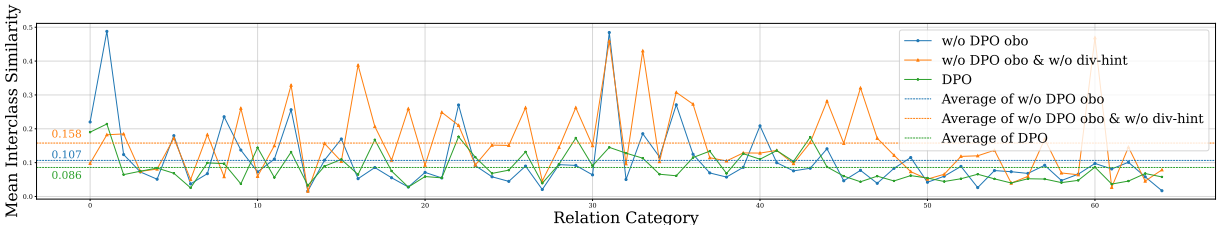

Figure 5: Average cosine similarity between generated training samples (K=32) for each relation category.

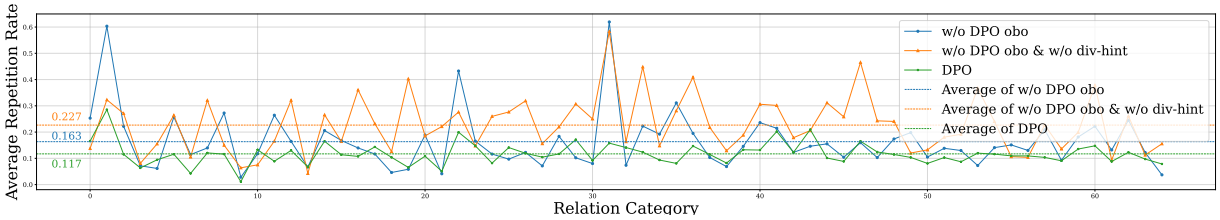

Figure 6: Average repetition rate of words between generated training samples (K=32) for each relation category.

relation extraction training samples generated by the LLMs after DPO fine-tuning training is lower than that of the two cases of LLMs w/o DPO, which again illustrates the effectiveness of the DPO fine-tuning in the task of generating training samples. Furthermore, the instruction on "diversity" in the Prompt of the LLMs *"Please make the generated samples as different from the above demonstrations as possible."* also plays an important role, because the overall average diversity of the generated training samples is 0.051 lower than without this instruction.

To further analyze the diversity, we also calculate the average repetition rate of words between every two pairs of the generated samples, as shown in Figure 6. It is apparent that the repetition rate of words of training samples generated after DPO fine-tuning is lower than both of other cases.

5.4 Number of Generated Samples

The generation of more training samples does not always improve the performance of the KnowPrompt, which has an upper limit. We generated 8, 16, 32 and 64 samples on four relation extraction datasets using LLMs with ICL and DPO. From the results in Figure 7, we find that the performance of the KnowPrompt increases and then remains constant as the number of samples generated by the LLMs increases, while peaking near K = 32. We argue that it is because the high-diversity of training samples generated by LLMs reaches the end point near K = 32 due to the limitation of the corpus training database of LLMs. It is also for this reason that the research in this paper is centered around 8-shot, 16-shot and 32-shot based.

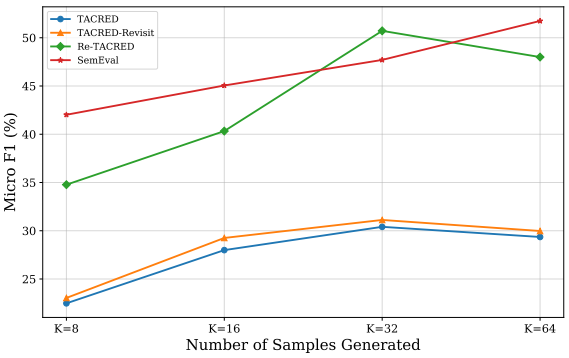

Figure 7: Micro F1 (%) of KnowPrompt using different numbers of training samples generated by LLMs.

5.5 Case Study

In response to the experimental phenomena in §5.3 Diversity between Training Samples and §5.4 Number of Generated Samples, we develop a specific case study using generated training samples on the "per:country_of_death" relation of TACRED.

As shown in Table 4, our method allows LLMs to generate training samples with low similarity, and to maintain diversity in the verbalized representation of relations. On the contrary, after removing the "diversity instruction" and DPO, the generated training samples have higher similarity and single verbalized representation of relations, and even the post-order samples appear to be exactly the same as the pre-order samples. Meanwhile, we find that

Method	Step	Generated Data
Ours	1	John Smith$_{subj}$ died in Canada$_{obj}$ after a long illness and was buried there.
	4	Frank Ocean$_{subj}$ ended his life **in** France$_{obj}$ where he had lived for many years.
	6	Bob Smith$_{subj}$ expired **in** China$_{obj}$ **where he** worked **for** decades.
	9	Angela Davis$_{subj}$ passed away **in** Sweden$_{obj}$ **where** she fought **for** civil rights.
	13	James Bond$_{subj}$ **ended his life in** Monaco$_{obj}$ **where he** found himself unwanted.
	16	Tupac Shakur$_{subj}$ **passed away in** Grenada$_{obj}$.
Ours, w/o DI, w/o DPO	1	France$_{obj}$ John F Kennedy$_{subj}$ American Kennedy family Assassination Dallas died.
	4	China$_{obj}$ Wang Zhiyuan$_{subj}$ computer scientist **died** Beijing age 62.
	6	South Africa$_{obj}$ Nelson Mandela$_{subj}$ **died** Johannesburg **age** 95.
	9	Germany$_{obj}$ Heinrich Himmler$_{subj}$ Nazi official **died** Berlin **age** 44.
	13	**China Wang Zhiyuan computer scientist died Beijing age 62.**
	16	Canada$_{obj}$ Stephen Harper$_{subj}$ politician **died** Ottawa **age** 68.

Table 4: A case study of step-by-step generation of training samples on the "per:country_of_death" relation of TACRED. We mark in **red** the content of the training samples generated in the post-order that are identical to the content of the pre-order.

although our method enables LLMs to generate training samples with high diversity, the training samples generated by LLMs around Step = 16 also appear to be similar in the verbalized representations of relations, indicating that the similarity of the training samples generated later will increase. This results in the performance of the KnowPrompt peaking shortly after K = 16 and remaining largely unchanged thereafter.

6 Conclusion

In this paper, we propose a method for generating training samples for RE with LLMs. The method optimizes the output of LLMs to generate high-quality training samples for RE, especially in terms of diversity, through ICL and DPO. ICL allows LLMs to quickly learn the structure and content of training samples by providing appropriate sample demonstrations in the prompt; DPO allows LLMs to generate training samples with both diversity and correctness in mind through fine-tuning. Experiments demonstrate the effectiveness of these generated training samples in few-shot scenarios, especially with greater advantages in diversity.

Limitations

Despite our best efforts, the method proposed in this paper may still have some limitations.

LLMs: Although we have enabled the LLMs to generate better quality training samples by fine-tuning the training, the quality of these training samples is also largely limited by the strength of the open-source LLMs themselves.

Maximum Number: The maximum number of valid training samples that can be generated by LLMs is very limited, as performance does not consistently improve after generating about 16 or 32 training samples.

Acknowledgements

This research is supported by the National Natural Science Foundation of China (No. 62306140, No. 62476127), the Natural Science Foundation of Jiangsu Province (No. BK20242039), the Basic Research Program of the Bureau of Science and Technology (ILF24001), the Fundamental Research Funds for the Central Universities (No. NJ2023032), the Scientific Research Starting Foundation of Nanjing University of Aeronautics and Astronautics (No. YQR21022), the Key Project

of Jiangsu Collaborative Innovation Center of Chinese Medicinal Resources Industrialization (No. 000003401025-6), the Open Project of Chinese Materia Medica First-Class Discipline of Nanjing University of Chinese Medicine (No. ZYXJC2024-010) and the High Performance Computing Platform of Nanjing University of Aeronautics and Astronautics.

References

Christoph Alt, Aleksandra Gabryszak, and Leonhard Hennig. 2020. TACRED revisited: A thorough evaluation of the TACRED relation extraction task. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 1558–1569. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Xiang Chen, Lei Li, Ningyu Zhang, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. 2022a. Relation extraction as open-book examination: Retrieval-enhanced prompt tuning. In *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*, pages 2443–2448. ACM.

Xiang Chen, Ningyu Zhang, Xin Xie, Shumin Deng, Yunzhi Yao, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. 2022b. Knowprompt: Knowledge-aware prompt-tuning with synergistic optimization for relation extraction. In *WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022*, pages 2778–2788. ACM.

Yew Ken Chia, Lidong Bing, Soujanya Poria, and Luo Si. 2022. Relationprompt: Leveraging prompts to generate synthetic data for zero-shot relation triplet extraction. In *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 45–57. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, Lei Li, and Zhifang Sui. 2023. A survey for in-context learning. *CoRR*, abs/2301.00234.

Jiahui Gao, Renjie Pi, Yong Lin, Hang Xu, Jiacheng Ye, Zhiyong Wu, Weizhong Zhang, Xiaodan Liang, Zhenguo Li, and Lingpeng Kong. 2023. Self-guided noise-free data generation for efficient zero-shot learning. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Zhijiang Guo, Guoshun Nan, Wei Lu, and Shay B. Cohen. 2020. Learning latent forests for medical relation extraction. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 3651–3657. ijcai.org.

Zhijiang Guo, Yan Zhang, and Wei Lu. 2019. Attention guided graph convolutional networks for relation extraction. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 241–251. Association for Computational Linguistics.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2010. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the 5th International Workshop on Semantic Evaluation, SemEval@ACL 2010, Uppsala University, Uppsala, Sweden, July 15-16, 2010*, pages 33–38. The Association for Computer Linguistics.

Guozheng Li, Peng Wang, and Wenjun Ke. 2023. Revisiting large language models as zero-shot relation extractors. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 6877–6892. Association for Computational Linguistics.

Yang Liu, Jinpeng Hu, Xiang Wan, and Tsung-Hui Chang. 2022. Learn from relation information: Towards prototype representation rectification for few-shot relation extraction. In *Findings of the Association for Computational Linguistics: NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 1822–1831. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis,

Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Yu Meng, Jiaxin Huang, Yu Zhang, and Jiawei Han. 2022. Generating training data with language models: Towards zero-shot language understanding. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

OpenAI. 2022. Chatgpt: Optimizing language models for dialogue.

OpenAI. 2023. Gpt-4 technical report. arXiv preprint, abs/2303.08774.

Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, Rishita Anubhai, Cícero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. 2021. Structured prediction as translation between augmented natural languages. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Meng Qu, Tianyu Gao, Louis-Pascal A. C. Xhonneux, and Jian Tang. 2020. Few-shot relation extraction via bayesian meta-learning on relation graphs. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 7867–7876. PMLR.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Rohini K. Srihari and Wei Li. 2000. A question answering system supported by information extraction. In *6th Applied Natural Language Processing Conference, ANLP 2000, Seattle, Washington, USA, April 29 - May 4, 2000*, pages 166–172. ACL.

George Stoica, Emmanouil Antonios Platanios, and Barnabás Póczos. 2021. Re-tacred: Addressing shortcomings of the TACRED dataset. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 13843–13850. AAAI Press.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Zhen Wan, Fei Cheng, Zhuoyuan Mao, Qianying Liu, Haiyue Song, Jiwei Li, and Sadao Kurohashi. 2023. GPT-RE: in-context learning for relation extraction using large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 3534–3547. Association for Computational Linguistics.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 13484–13508. Association for Computational Linguistics.

Xin Xu, Yuqi Zhu, Xiaohan Wang, and Ningyu Zhang. 2023. How to unleash the power of large language models for few-shot relation extraction? In *Proceedings of The Fourth Workshop on Simple and Efficient Natural Language Processing, SustaiNLP 2023, Toronto, Canada (Hybrid), July 13, 2023*, pages 190–200. Association for Computational Linguistics.

Jiacheng Ye, Jiahui Gao, Qintong Li, Hang Xu, Jiangtao Feng, Zhiyong Wu, Tao Yu, and Lingpeng Kong. 2022. Zerogen: Efficient zero-shot learning via dataset generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 11653–11669. Association for Computational Linguistics.

Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 1753–1762. The Association for Computational Linguistics.

Kai Zhang, Bernal Jimenez Gutierrez, and Yu Su. 2023. Aligning instruction tasks unlocks large language

models as zero-shot relation extractors. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 794–812. Association for Computational Linguistics.

Yuhao Zhang, Peng Qi, and Christopher D. Manning. 2018. Graph convolution over pruned dependency trees improves relation extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2205–2215. Association for Computational Linguistics.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 35–45. Association for Computational Linguistics.

Lingfeng Zhong, Jia Wu, Qian Li, Hao Peng, and Xindong Wu. 2024. A comprehensive survey on automatic knowledge graph construction. *ACM Comput. Surv.*, 56(4):94:1–94:62.

Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 2: Short Papers*. The Association for Computer Linguistics.

Wenxuan Zhou and Muhao Chen. 2022. An improved baseline for sentence-level relation extraction. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing, AACL/IJCNLP 2022 - Volume 2: Short Papers, Online only, November 20-23, 2022*, pages 161–168. Association for Computational Linguistics.

A Experimental Details

A.1 Datasets

For comprehensive experiments, we conducted experiments on four relation extraction datasets: TACRED, TACRED-Revisit, Re-TACRED and SemEval 2010 Task 8 (SemEval). A brief introduction to these data is given below:

TACRED: a large-scale sentence-level relation extraction dataset from the annual TACBP4 challenge, containing over 106,000 sentences. It involves 42 different relation categories, including 41 common relation categories and a special "no relation" relation category.

TACRED-Revisit: a dataset constructed on the basis of the TACRED dataset. The researchers found errors in the development and test sets of the original TACRED dataset and corrected them while keeping the training set intact.

Re-TACRED: another version of the TACRED dataset, which addresses some of the shortcomings of the original TACRED dataset by reconstructing the training, validation and test sets. Meanwhile, this dataset removes the original 6 relation categories and adds 4 new relation categories to the TACRED dataset, so that a dataset with 40 relation categories is finally obtained.

SemEval: a traditional relation extraction dataset, containing 10,717 annotated samples, covers 9 bi-directional relation categories and a special "no relation" relationship category.

A.2 Relation Explanation

We give explanations for each relation in the four datasets. The detailed explanation for each relation is shown in Table 5.

Relation	Explanation
Component-Whole (e2,e1)	Tail entity e2 is the component of head entity e1, and head entity e1 is the whole of tail entity e2
Instrument-Agency (e2,e1)	Tail entity e2 is the instrument of head entity e1, and head entity e1 is the agency of tail entity e2
Member-Collection (e1,e2)	Head entity e1 is the member of tail entity e2, and tail entity e2 is the collection of head entity e1
Cause-Effect (e2,e1)	Tail entity e2 is the cause of head entity e1, and head entity e1 is the effect of tail entity e2
Entity-Destination (e1,e2)	Head entity e1 is the entity of tail entity e2, and tail entity e2 is the destination of head entity e1
Content-Container (e1,e2)	Head entity e1 is the content of tail entity e2, and tail entity e2 is the container of head entity e1
Message-Topic (e1,e2)	Head entity e1 is the message of tail entity e2, and tail entity e2 is the topic of head entity e1
Product-Producer (e2,e1)	Tail entity e2 is the product of head entity e1, and head entity e1 is the producer of tail entity e2
Member-Collection (e2,e1)	Tail entity e2 is the member of head entity e1, and head entity e1 is the collection of tail entity e2
Entity-Origin (e1,e2)	Head entity e1 is the entity of tail entity e2, and tail entity e2 is the origin of head entity e1
Cause-Effect (e1,e2)	Head entity e1 is the cause of tail entity e2, and tail entity e2 is the effect of head entity e1
Component-Whole (e1,e2)	Head entity e1 is the component of tail entity e2, and tail entity e2 is the whole of head entity e1
Message-Topic (e2,e1)	Tail entity e2 is the message of head entity e1, and head entity e1 is the topic of tail entity e2
Product-Producer (e1,e2)	Head entity e1 is the product of tail entity e2, and tail entity e2 is the producer of head entity e1
Entity-Origin (e2,e1)	Tail entity e2 is the entity of head entity e1, and head entity e1 is the origin of tail entity e2
Content-Container (e2,e1)	Tail entity e2 is the content of head entity e1, and head entity e1 is the container of tail entity e2
Instrument-Agency (e1,e2)	Head entity e1 is the instrument of tail entity e2, and tail entity e2 is the agency of head entity e1
Entity-Destination (e2,e1)	Tail entity e2 is the entity of head entity e1, and head entity e1 is the destination of tail entity e2
Other	Tail entity e2 is the component of head entity e1, and head entity e1 is the whole of tail entity e2
org:founded	The founding relationship of an organization
org:subsidiaries	The subsidiaries of an organization
per:date_of_birth	The date of birth of a person
per:cause_of_death	The cause of death of a person
per:age	The age of a person
per:stateorprovince_of_birth	The state or province of birth of a person
per:countries_of_residence	The countries where a person resides

Relation	Explanation
per:country_of_birth	The country of birth of a person
per:stateorprovinces_of_residence	The states or provinces where a person resides
org:website	The website of an organization
per:cities_of_residence	The cities where a person resides
per:parents	The parents of a person
per:employee_of	The organization where a person is employed
NA/no_relation	Unknown relation
per:city_of_birth	The city of birth of a person
org:parents	The parent company of an organization
org:political/religious_affiliation	The political or religious affiliation of an organization
per:schools_attended	The schools attended by a person
per:country_of_death	The country where a person died
per:children	The children of a person
org:top_members/employees	The top members/employees of an organization
per:date_of_death	The date of death of a person
org:members	The members of an organization
org:alternate_names	The alternate names of an organization
per:religion	The religion of a person
org:member_of	The organization to which a member belongs
org:city_of_headquarters	The city where the headquarters of an organization is located
per:origin	The origin of a person
org:shareholders	The shareholders of an organization
per:charges	The charges against a person
per:title	The title of a person
org:number_of_employees/members	The number of employees/members in an organization
org:dissolved	The date of dissolution of the organization
org:country_of_headquarters	The country where headquarters of an organization is located
per:alternate_names	The alternate names of a person
per:siblings	The siblings of a person
org:stateorprovince_of_headquarters	The state or province where headquarters of an organization is located
per:spouse	The spouse of a person
per:other_family	Other family members of a person
per:city_of_death	The city where a person died
per:stateorprovince_of_death	The state or province where a person died
org:founded_by	The founder of an organization
org:country_of_branch	The country where a branch of an organization is located
org:city_of_branch	The city where a branch of an organization is located
org:stateorprovince_of_branch	The state or province where branch of an organization is located
per:identity	The identity information or characteristics of a person

Table 5: Explanation of each relation in the four datasets.