Enhancing Zero-Shot Relation Triplet Extraction through Staged Interaction with Large Language Models

Anonymous ACL submission

Abstract

Zero-shot Relation Triplet Extraction (ZeroRTE) is a challenging yet valuable task that extracts relation triplets from unstructured texts for new relation types, significantly reducing the time and effort needed for data labeling. With the enhancement of the zero-shot capability of large language model, the performance of many zero-shot tasks has been further improved only via chatting with Large Language Model(LLM). In this work, we transform the zero-shot triplet extraction task into a two-stage chat with LLM. Specially, followed by the step of triplet extraction, we prompt the LLM to perform the NER(Name Entity Recognition) task in the first stage. Then, in the second stage, we prompt the LLM to perform the RC(Relation Classification) task combining the result of the first stage. To overcome the impact of redundant information of the LLM's output on task evaluation, we design a Post-Processing module to obtain the relation triplet. Experiments on Wiki-ZSL and FewRel datasets show the efficacy of Relation Prompt for the ZeroRTE task. Remarkably, our method outperforms strong baselines by a significant margin, achieving an impressive 15.89% increase in F1 scores, particularly when dealing with Wiki-ZSL with 15 unseen relations.

1 Introduction

Relation Triplet Extraction aims to extract a full triplet, namely(head_entity, relation_type, tail_entity) for an unstructured text, which has applications such as knowledge graph construction and question answering(Wadhwa et al., 2023). However, existing approaches often require large datasets of annotated samples which are costly to annotate and have a fixed set of relations. Hence, many researchers have been aware of IE techniques with zero/few-shot methods. Under the zero-shot setting, the relation sets at the training and testing stages are disjoint. ZeroRTE models are trained on samples with a handful of seen relation types and are expected to generalize to extract triplets with previously unseen relation types.

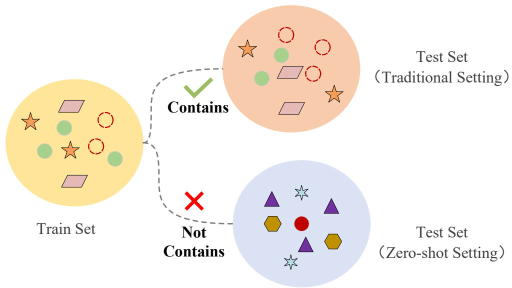

Figure 1: Zero-shot relation triplet extraction.

With the widespread application of Large language models, their zero-shot capabilities have also received increasing attention.(Brown et al., 2020) Using carefully designed prompts has become a popular way to unleash the potential of large models on zero-shot tasks(Bi et al., 2024). The Chain of Thought (CoT) prompting(Wei et al., 2022), which involves gradually prompting LLM to break down complex problems into step-by-step sub-problems, has been proven effective for LLM in solving complex issues. Though it is non-trivial to tackle ZeroRTE by decomposing it into two sub-tasks, **How to plan the two-stage tasks?** and **How to design the prompt to enhance the ZRTE ability of LLM?** are still the challenges that need to be addressed. RSED(Lan et al., 2024) and ChatIE(Wei et al., 2023) decomposed ZeroRTE into Relation Selection and Entity boundary Detection, which will increase the propagation of errors, as the first stage task Relation Selection is a task that requires a deeper understanding of semantics compared to NER.

In our work, ZeroRTE is divided into two uncomplicated subtasks, NER and Relation Classification. By decomposing the complex ZeroRTE

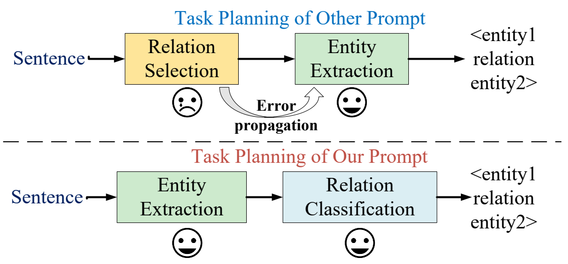

Figure 2: Task Planning of different prompt

task into two simpler tasks than Relation Selection and Entity boundary Detection, the error propagation is alleviated, which is more conducive for the model to solve the RTE. For each stage, we consider designing the prompt from three aspects: 1)Task Description: Concisely summarize the task objectives and requirements. 2)Background knowledge: the background information the LLM should be combined with, such as the results of the first stage task NER. 3)Output Format: Clearly define the expected format of the output data to meet the task requirements.

Following the method above, extensive experiments are conducted on the ZeroRTE tasks. Specifically, our method advances the state-of-the-art RSED(Lan et al., 2024)model on two ZeroRTE datasets and gains an improvement of up to 15.89% in the F1 score over the previous best model on Wiki-ZSL and FewRel. Compared with the previous ZeroRTE method, our method has obvious advantages. The previous method required pretraining the model and generating synthetic data, but our approach only requires Parameter-Efficient Fine-Tuning without the need for synthetic data.

The main contributions of this paper are summarized as follows:

(1)We introduced an innovative two-stage framework leveraging LLM for zero-shot relation triplet extraction, highlighting our innovative use of NER and relation classification tasks, enhanced by a carefully designed prompt, which significantly reduces error propagation.

(2)We compared the impact of different two-stage task planning and various prompt designs on the ZeroRTE task and conducted a detailed analysis.

(3)Experiments on two datasets demonstrate that our proposed method is state-of-the-art (SOTA) method in the field of ZeroRTE.

2 Related Works

2.1 Zero-shot Relation Triplet Extraction

RelationPrompt(Chia et al., 2022) first formally introduced the task setting of Zero-shot Relation Extraction (ZeroRTE), utilizing synthetic data generated by prompting language models to generate structured texts. ChatIE(Wei et al., 2023), for the first time, utilizes LLM to address the problem of zero-shot information extraction through multi-turn dialogues with ChatGPT[1]. It consists of two stages: category selection and relationship generation. While both category selection and relationship generation are a difficult task that need a deeper understanding of semantics. Then, the utilization of ChatGPT incurs significant financial costs and may also be inaccessible in certain geographical regions.

2.2 Prompt Engineering

Prompting-based methods have shown promise as a new paradigm for zero-shot or few-shot inference in natural language processing. Recent progress in LLM prompt-tuning aims to bridge the gap between pre-training and downstream tasks by using natural language templates.(Pourpanah et al., 2022) Chain of thought (CoT) prompting(Wei et al., 2022) , an instance of few-shot prompting, proposed a simple solution by modifying the answers in few-shot examples to step-by-step answers, and achieved significant boosts in performance across these difficult benchmarks, especially when combined with very large language models. ZETT(Kim et al., 2023) tackles the Zero-shot Triplet extraction by Template infilling. This method designed the template for each relation, which is not feasible for situations with a large number of relationships. ChatIE(Wei et al., 2023) used the Chat-based Prompt to tackle the Zero-shot Information extraction.

3 Method

3.1 Task Formulation

Let $D = \{S, < head_entity, R, tail_entity >\}$denotes the whole dataset, consisting of the input sentences S, the output triplets $< head\_entity, R, tail\_entity >$ where R is the set of relation labels. $D = D_s \cup D_u$, where D_s, D_u refer to the seen and unseen datasets respectively. The model is trained on D_s and evaluated on D_u.

[1] https://openai.com/chatgpt/

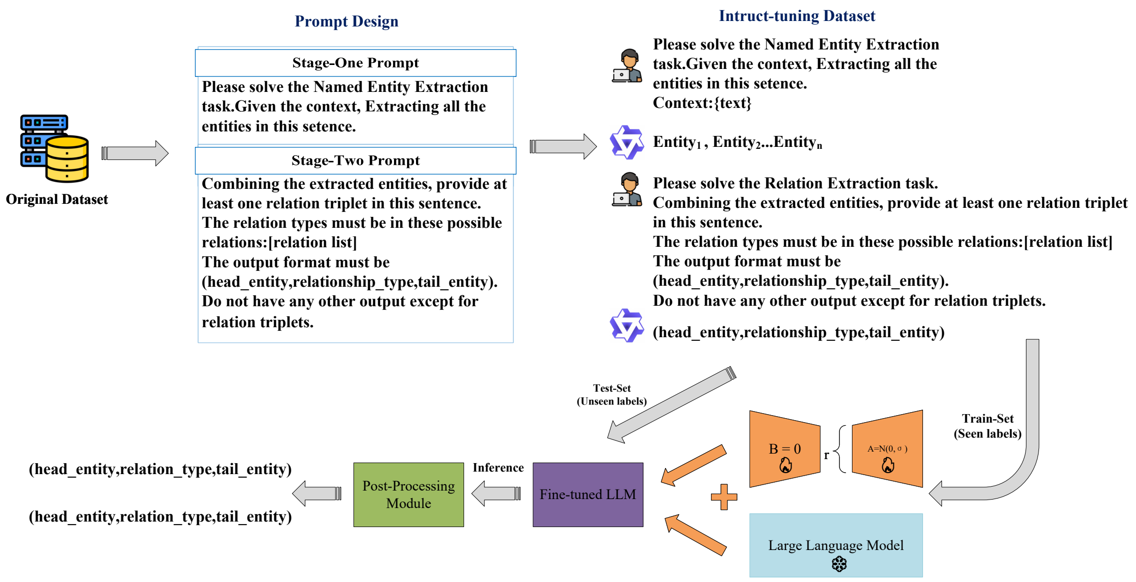

Figure 3: Overall Architecture of Models.

$R = R_s \cup R_u$ is predefined, comprising the seen relation label set $R_s = \{r_1^s, \cdots, r_n^s\}$ and unseen relation label set $R_u = \{r_1^u, \cdots, r_m^u\}$, where $n = |R_s|$ and $m = |R_u|$ are the number of seen and unseen relation labels respectively. $R_s \cap R_u = \emptyset$, R_s and R_u are disjoint. One sentence $s \in S$ contains one or more triplets.

3.2 Overview

To tackle the ZeroRTE task, we proposed a two-stage dialogue framework based on a Large Language Model (LLM). The framework initially divides the task into two phases: Named Entity Recognition (NER) and Relation Classification (RC), followed by the careful design of prompts for each stage. Based on the carefully designed prompts, we have formulated an instruct-tuning dataset. Then, fine-tuning the LLM with the Low-Rank Adaptation (LoRA) method(Hu et al., 2021) based on the instruct-tuning dataset. Subsequently, the fine-tuned model is applied to infer on a test set that encompasses unseen labels. The inference results are then refined by the post-processing module to obtain the definitive set of relation triplets.

3.3 Two-stage Prompts Design

Prompt engineering has emerged as a crucial technique for enhancing the capabilities of pre-trained large language models (LLMs)(Liu et al., 2023). The significance of prompt engineering is especially evident in its transformative impact on the adaptability of LLMs.

We have designed the prompt for zero-shot tasks in two stages, aiming to break down the complex task of relation triplet extraction into two simpler tasks to enhance the performance of large models in relation extraction tasks. The tasks for the first and second stages are NER (Named Entity Recognition) and RC (Relation Classification), respectively. Below, we will provide a detailed introduction to the specific design of the prompts for both stages. The overall architecture of prompt design is presented in the Figure 4.

3.3.1 Stage-One: Named Entity Recognition

ChatIE(Wei et al., 2023) and RSED(Lan et al., 2024) transform the Relation Triplet Extraction into relation category selection and relation generation. In our experiment, it reveals that LLM is prone to significant bias during the relation type selection phase when using this method, which in turn affects the final performance of relation extraction. This is because the selection of relation types requires a deep understanding of semantics to make the correct choices. In contrast, the task of entity recognition is a relatively simpler task that only requires identifying the relevant entities in the text.

Therefore, in our research, we have replaced the relation type selection task with entity recognition and have designed the following prompts:*"Please solve the Named Entity Extraction task.the context, Extracting all the entities in this sentence.Context:{text}"*, aimed at providing more

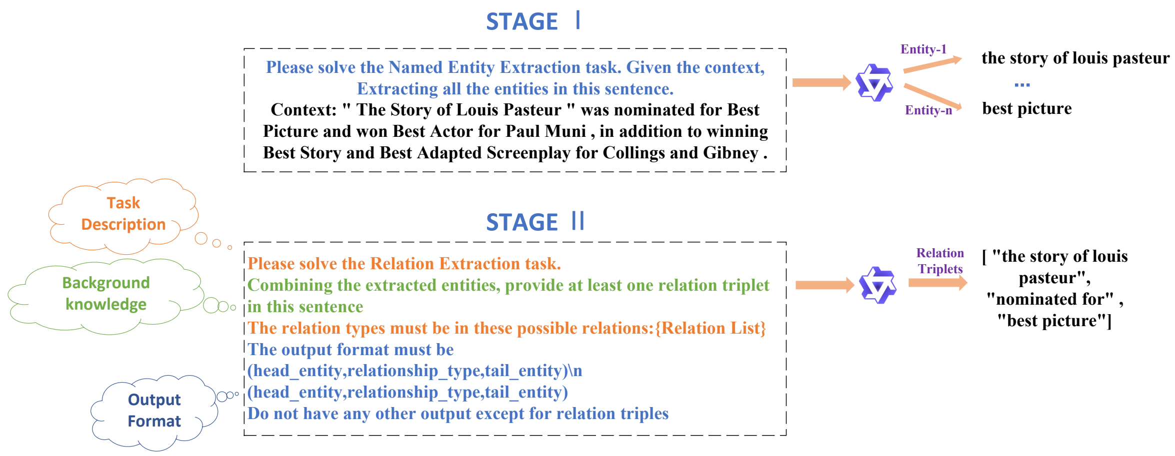

Figure 4: Two-Stage Prompt Design.

comprehensive information to support subsequent tasks. By applying this prompt, we require the LLM to generate all the entities contained in the text, avoiding the absence of entity information.

3.3.2 Stage-Two: Relation Classification and Triplets Generation

This stage is the core phase of relation triplet extraction, responsible for combining the results of the previous entity extraction round, performing relationship classification, and generating triplets that conform to the specified format. The design of prompt in this stage is mainly considered from three aspects:

(1)Task Description: In this stage, the prompt is designed to clearly specify the task of relation triplet extraction. The prompt should guide the model to generate triples that adhere to the specified format.

(2)Background Information: It should provide instructions on how to combine the entities extracted in the previous round with the provided context. Perform relationship classification based on this information and generate relation triplets in the specified format. Ensure that each triplet consists of a subject entity, a relationship label, and an object entity.

(3)Output Format: The prompt should clearly define the expected output format for the generated relation triplets. It should specify the structure and organization of each triplet, including the order of entities and the representation of the relationship label. This ensures that the model produces output that conforms to the desired format.

Following the three design principles mentioned above, we have designed the prompt for the second stage as follows:*"Please solve the Relation Extraction task. Combining the extracted entities, provide at least one relation triplet in this sentence. The relation types must be in these possible relations:[relation list]The output format must be (head_entity,relationship_type,tail_entity). Do not have any other output except for relation triplets."*

After the design of the prompt is completed, the existing dataset is transformed to form an instruction dataset.

$$\mathcal{D}_{instruct} = Prompt(\mathcal{D}_{origin}) \tag{1}$$

Here, $Prompt(\cdot)$ denotes the method of transfering the original datset to the instruct-tuing datset combining our designed prompt.

3.4 LLM Fine-tuning

After finalizing the prompt design, we combined it with the dataset to create the instruct-tuning dataset for our work. Subsequently, based on the instruct-tuning dataset, we fine-tune (LoRA) the LLM.

There are two training tasks in our model: named entity recognition and relation triplets generation. We train the model in a multi-task manner. For named entity recognition, the training objective is to minimize the cross entropy loss:

$$\mathcal{L}_{ent} = -\sum_{c=1}^{C} y_{o,c} \log(p_{o,c}) \tag{2}$$

For relation triplets generation, we also adopt the cross entropy loss, and the entity loss is defined as:

$$\mathcal{L}_{rel} = -\sum_{t=1}^{T} y_{o,t} \log(p_{o,t}) \tag{3}$$

Here, $\mathcal{L}_{ent}$ denotes the loss function of the NER task. $\mathcal{L}_{rel}$ denotes the loss function of the relation triplets generation task. C represents the number of tokens for the entity to be predicted. $y_{o,c}$is the true label (usually token id) for c_{th} in the first stage, $p_{o,c}$is the predicted probability for class in the first stage. T represents the number of tokens for the triplets to be predicted. $y_{o,t}$is the true label (usually token id) for t_{th} in the first stage, $p_{o,t}$is the predicted probability for class in the first stage. We treat each loss equally and the model learns to minimize $\mathcal{L} = \mathcal{L}_{rel} + \mathcal{L}_{ent}$ jointly.

Algorithm 1: Post-Processing Method

```
Data: Model 𝓕;
  Test dataset 𝓓;
  Relation List in the test set 𝓡;
  Number of samples in the test set 𝓝;
Result: Relation Triplets
  𝓣(head_entity_i, rel, tail_entity_i)
for m = 1, 2, ..., 𝓝 do
    x_m ← Get(𝓓, m);
    y_m ← Normalize(𝓕(x_m));
    S_{y_m} ← Split(y_m);
    𝓝_S ← Number(S_{y_m});
    for q = 1, 2, ..., 𝓝_S do
        if len(S_{y_m,q}) = 3 then
            if get(S_{y_m,q}, 1) ∈ 𝓡 then
                h_m = get(S_{y_m,q}, 0);
                r_m = get(S_{y_m,q}, 1);
                t_m = get(S_{y_m,q}, 2);
    t ← (h_m, r_m, t_m);
    Add t To 𝓣;
return 𝓣
```

3.5 Post-Processing Module

Because the LLMs are generative models, their outputs may contain additional descriptive information beyond the relation triplets. To address this, we have developed a post-processing module that purifies the content generated by the LLM and only extracts the triplets we need. Moreover, it filters out triples that do not match the predefined relation type list, thereby enhancing the accuracy and relevance of our model's output results. The details as follows Algorithm 1.

After inputting each test sample x_m from the instruct-tuning dataset into the model, we obtain the model's feedback output $\mathcal{F}(x_m)$. We then proceed with our post-processing procedure, which first involves standardizing the model output. This mainly includes unifying punctuation marks, such as replacing Chinese commas with English commas. Next, we split the standardized output y_m, which includes dividing each line based on line breaks and splitting the elements contained in each line based on commas. Then, we determine if the number of elements per line is 3. Only when it is equal to 3 can we assert that the content of this line represents a relation triplet. Finally, we filter out triplets that are not in the predefined relation list. Eventually, the final relation triplets are formed.

4 Experiments

4.1 Experimental Settings

For the task of Zero-shot relation triplets extraction, we fine-tune the Qwen1.5-14B-Chat[2] which has 14B parameters. A machine equipped with 8x NVIDIA GeForce RTX 3090 GPUs (each with 24GB of VRAM) is used for training. The fine-tuning is performed on the training set for up to 10 epochs using LoRA method. The learning rate is 1e-4 with linear warm-up for the first 100 training steps and the batch size is set to 4. During the training process, we use the AdamW optimizer.

Since generative models cannot determine the number of triplets they generate, we evaluate the triplet extraction results at the multiple triplets setting. To evaluate multiple triplet extraction, we use the Micro F1 metric which is standard in structured prediction tasks, and report the precision (P.) and recall (R.)

4.2 Datasets

We use the following two datasets for our experiments. FewRel(Han et al., 2018) was hand-annotated for few-shot relation extraction, but we made it suitable for the zero-shot setting after data splitting into disjoint relation label sets for training and testing. Wiki-ZSL (Chen and Li, 2021) is constructed through distant supervision over Wikipedia articles and the Wikidata knowledge base. For each dataset, we set the unseen label size to $m \in \{5, 10, 15\}$, while treating the remaining labels as seen labels during training in the experiments. The detailed statistics of the Dataset are shown in Table 2.

[2]Qwen refers to the large language model family built by Alibaba Cloud. https://huggingface.co/Qwen

Table 1: Results Compared with baseline models(Training Mode)

Unseen Labels	Method	Wiki-ZSL			Fewrel		
		P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)
m=15	TableSequence(Wang and Lu, 2020)	44.43	3.53	6.39	19.03	1.99	3.48
	RelationPrompt(Chia et al., 2022)	26.19	32.12	28.85	17.73	23.20	20.08
	PAED(Zhu et al., 2023)	-	-	-	20.68	23.39	21.95
	RSED(Lan et al., 2024)	25.37	33.80	28.98	27.00	23.55	25.16
	Ours	**57.31**	**36.87**	**44.87**	**30.28**	**25.03**	**27.41**
m=10	TableSequence(Wang and Lu, 2020)	45.31	3.57	6.4	28.93	3.60	6.37
	RelationPrompt(Chia et al., 2022)	30.20	32.31	31.19	21.59	28.68	24.61
	PAED(Zhu et al., 2023)	-	-	-	23.31	27.42	25.15
	RSED(Lan et al., 2024)	27.09	**39.09**	32.00	30.89	29.90	30.39
	Ours	**60.77**	37.76	**46.57**	**35.46**	**32.17**	**33.74**
m=5	TableSequence(Wang and Lu, 2020)	43.68	3.51	6.29	15.23	1.91	3.40
	RelationPrompt(Chia et al., 2022)	29.11	31.00	30.01	20.80	24.32	22.34
	PAED(Zhu et al., 2023)	-	-	-	25.79	34.54	29.47
	RSED(Lan et al., 2024)	38.14	**36.84**	**37.48**	43.91	34.97	38.93
	Ours	**42.52**	23.27	30.09	**53.10**	**52.66**	**52.88**

Table 2: Statistics of Datasets

Dataset	Samples	Entities	Relation Labels			Average Length
			Total	Train	Test	
Wiki-ZSL	94383	77623	113	98	5	24.85
				98	10	
				98	15	
FewRel	56000	72954	80	65	5	24.95
				65	10	
				65	15	

4.3 Experimental Results

4.3.1 Experiments in Training Mode

Baseline Methods To demonstrate the effectiveness of our proposed method in the ZeroRTE task, we compared it with several existing ZeroRTE models.

- **TableSequence**(Wang and Lu, 2020) is a typical table-based method, which comprises two encoders to encode different types of information in the learning process.
- **RelationPrompt**(Chia et al., 2022) fine-tuned the BART(Lewis et al., 2020) on the synthetic relation sentences generated by prompting language models.
- **PAED**(Zhu et al., 2023) presented a generation-based framework for zero-shot PAED. A novel HNS strategy and a Meta-VAE sampler with CSC are presented to enhance the performance of this model.
- **RSED**(Lan et al., 2024) proposed a method with potential candidate relation selection and entity boundary detection directly utilizing the semantics of unseen relations to tackle the ZeroRTE task.

We report experimental results in Table 1. It can be seen that the performance of the proposed model in this paper is optimal in terms of P, R, and F1 values. We observe that our framework performs better when the number of unseen labels is 15. For example, our framework enhances F1 index by 2.25% on the Fewrel dataset and 15.98% on the Wiki-ZSL dataset compared to the RSED which is the previous SOTA method. In conclusion, we demonstrate that our proposed method can better unleash the potential of large models in the RTE task.

However, it cannot be denied that when the number of unseen labels is 5, our method's performance on the Wiki-ZSL dataset is not as good as the RSED method. This may be related to the distribution of the Wiki-ZSL dataset. What's more, according to our observations, when there are fewer labels, it is also easy to generate triplets outside of the predefined relations, which is also one of the reasons for its poor performance. Nevertheless, its performance still exceeds that of other models. This is also one of the areas we will focus on improving in the future.

4.3.2 Experiment in No Training Required Mode

Taking into account the zero-shot capabilities inherent in LLM, we performed an experiment that does not alter the model parameters with training data, but instead directly utilizes the LLM for inference under various prompts. This allows us to compare

Table 3: Results Compared with baseline model(No Training Required Mode).† denotes the results from TAG(Xu et al., 2024).

Method	BaseModel	Wiki-ZSL			Fewrel		
		P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)
ICL†	GPT-3.5	8.87	8.68	8.49	11.35	12.58	11.87
ChatIE†	GPT-3.5	8.52	8.01	8.15	11.11	10.93	10.99
RelationPrompt†	GPT-3.5	7.76	6.86	7.28	8.76	8.33	8.54
TAG + RelationPrompt†	GPT-3.5	10.08	8.50	9.21	11.75	10.98	11.35
Ours prompt	GPT-3.5	**19.31**	**11.04**	**14.05**	**28.91**	**15.72**	**20.37**

the superiority of our proposed two-stage prompt method over other prompt-based approaches.

Baseline Methods We compare our proposed Tow-stage prompt method, with competitive baselines in ZeroRTE.

• **ICL** is an in-context-learning method that directly prompts LLMs, we follow the prompting method in (Wei et al., 2023)

• **ChatIE** (Wei et al., 2023) transforms the ZeroRTE task into a multi-turn question-answering problem with a two-stage framework.

• **RelationPrompt(GPT-3.5)**(Chia et al., 2022) uses GPT-3.5 to generate synthetic data for unseen relations and then trains the extractor model BART on the synthetic data from GPT-3.5.

• **TAG+RelationPrompt(GPT-3.5)**(Xu et al., 2024) conducts the relation extraction through the interaction between two agents. One agent acts as a generator, using the same prompt as RelationPrompt to leverage GPT-3.5 for data generation. The other agent serves as an extractor, employing a BART model-based approach with reinforcement learning to perform triplet extraction.

The detail prompts for ICL, ChatIE, RelationPrompt and TAG can be seen in Appendix A.

We report experimental results in Table 3. Under the situation of no training required setting, the two-stage prompt method in this paper still performs much better than the baseline. Specifically, on the Wiki-ZSL dataset, our method achieves an F1 score that is 4.84% higher than the best previous method; on the FewRel dataset, it is 8.5% higher than the previous best method.

5 Analysis

5.1 Ablation Study

To validate the effectiveness of the proposed method in this paper, we conducted corresponding ablation experiments to assess the impact of each module on the model's performance.

Table 4: Ablation Study Results

Unseen Labels	Method	P(%)	R(%)	F1(%)
m=15	Ours ***w.o.first-stage prompt***	29.42	24.69	26.84
	Ours ***w.o.LoRA***	22.06	8.51	12.29
	Ours	**30.28**	**25.03**	**27.41**
m=10	Ours ***w.o.first-stage prompt***	33.30	26.93	29.78
	Ours ***w.o.LoRA***	17.18	4.76	7.45
	Ours	**35.46**	**32.17**	**33.74**
m=5	Ours ***w.o.first-stage prompt***	51.29	40.49	45.25
	Ours ***w.o.LoRA***	43.59	11.09	17.68
	Ours	**53.10**	**52.66**	**52.88**

We first removed the prompt from the first stage, using only the second stage's prompt and input text, which means only having a one-stage dialogue with the Fine-tuned model(trained on the two-stage prompt). The experiments showed that under different settings of the number of unseen labels, the model's performance experienced varying degrees of decline. It can be observed that the one-stage inference performs relatively worse than the two-stage inference model on the RTE task, indicating that the first stage can provide more information for the generation of the second stage's triples, thereby helping the model to generate better triples. However, the degree of decline is not particularly significant, with the decrease in the F1 score being only 0.57% when m=15. This also indirectly demonstrates that our model has good robustness. Even though the inference prompt is not completely the same as the training prompt, it can still maintain a certain level of performance.

Subsequently, we directly removed the fine-tuning module and utilized the model to perform inference solely based on the prompts we designed. The results indicate a significant drop in the model's performance. We can conclude that fine-tuning enhances the zero-shot recognition capability of LLM on the RTE task.

5.2 Comparison of Different prompt

To compare the impact of different prompts on model performance, we designed other two-stage

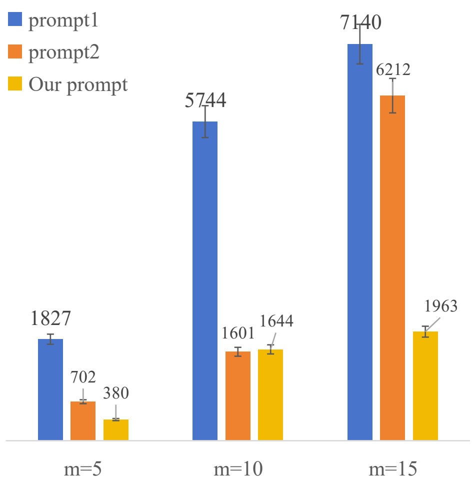

Figure 5: Number of Predictions Outside Predefined List.

models with different content for the experiments. Taking the FewRel as the test dataset, we conducted experiments under a different number of unseen label settings, and the experimental results are shown in Table 5.

• **prompt1** This prompt still adopts a two-stage task. Unlike the prompt used in this paper, we have changed the first stage to a relation type selection instead of the NER task.

• **prompt2** This prompt adopts the same two-stage task setup as this paper, but the way of expression is different.

The detail prompts for **prompt1** and **prompt2** can be seen in Appendix A.

Table 5: Results with different Prompt

Unseen Labels	Method	P(%)	R(%)	F1(%)
m=15	w.t. prompt1	**37.20**	11.91	18.04
	w.t. prompt2	17.73	23.20	20.08
	Ours prompt	30.28	**25.03**	**27.41**
m=10	w.t. prompt1	**51.59**	9.26	15.70
	w.t. prompt2	35.58	28.24	31.49
	Ours prompt	35.46	**32.17**	**33.74**
m=5	w.t. prompt1	**67.72**	32.37	43.80
	w.t. prompt2	50.85	41.09	45.45
	Ours prompt	53.10	**52.66**	**52.88**

The results of fine-tuning the large model with different prompts are shown in Table 5. The results indicate that using **prompt1**, which involves selecting relation types first and then generating triplets, performs poorly. From Figure 5, it can be seen that the number of predictions outside the predefined relation list is higher when using prompt1 for inference. This suggests that when using a two-stage approach for the relation triplet extraction task, the choice of tasks is extremely important.

The reasons for the poor performance of the Prompt1 task setup can be mainly attributed to two factors: 1)First, its initial phase of relation selection requires a deep understanding of semantics, which increases the difficulty of the task and contributes to the propagation of errors, resulting in a large number of incorrect relationship types or instances that exceed the predefined list. 2)Second, LLM has already been trained on a vast amount of data, acquiring certain capabilities in relation extraction. In most relation extraction tasks, the standard procedure is to first identify entities and then classify relations to form triplets. Therefore, LLM is more accustomed to and familiar with this task setup pattern. Hence, the method adopted in this paper, which involves entity recognition first followed by relation extraction, is more suitable for the Zero-RTE task.

At the same time, we used **prompt2**, which has the same task setup as this paper, but its expression is different. The main difference between prompt2 and the prompt used in this paper is that its task description is cumbersome, not as concise as the prompt used in this paper. Therefore, it can also be seen from the experimental results that the effect of using prompt2 is slightly inferior to the prompt used in this paper. Therefore, we can conclude that a more concise and clear prompt is more conducive to the LLM's understanding and completion of downstream tasks.

6 Conclusions and Future Work

This paper addresses the Zero-shot Relation Triplet Extraction (ZeroRTE) task, proposing an innovative two-stage conversational approach to enhance the capability of extracting relation triplets from unstructured texts for previously unseen relation types. The experimental results show that our method significantly outperforms strong baselines.

Future Work Our future work will focus on several aspects. Firstly, we will further optimize the Post-Processing module to more effectively address redundancy and noise in the LLM's output. Then, exploring different prompting strategies to enhance the performance of LLMs in zero-shot relation extraction tasks.

Limitation

We have demonstrated that across two standard ZeroRTE datasets, LLMs with our method achieve SOTA results. However, there are important limitations to these contributions. First, due to limitations in computational resources, we have only fine-tuned the Qwen LLM, and have not trained different types of large models. Once experimental conditions allow, we will conduct more comprehensive experiments on LLMs of different types and sizes. Then, our experiments are limited to datasets that are curated in English, so we cannot determine if the problems we have identified would manifest similarly in other languages.

References

Zhen Bi, Jing Chen, Yinuo Jiang, Feiyu Xiong, Wei Guo, Huajun Chen, and Ningyu Zhang. 2024. Codekgc: Code language model for generative knowledge graph construction. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 23(3):1–16.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Chih Yao Chen and Cheng-Te Li. 2021. Zs-bert: Towards zero-shot relation extraction with attribute representation learning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3470–3479.

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*, pages 45–57.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018. Fewrel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4803–4809.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2021. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Bosung Kim, Hayate Iso, Nikita Bhutani, Estevam Hruschka, Ndapandula Nakashole, and Tom Mitchell. 2023. Zero-shot triplet extraction by template infilling. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 272–284.

Yuquan Lan, Dongxu Li, Yunqi Zhang, Hui Zhao, and Gang Zhao. 2024. Rsed: Zero-shot relation triplet extraction via relation selection and entity boundary detection. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 11256–11260. IEEE.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35.

Farhad Pourpanah, Moloud Abdar, Yuxuan Luo, Xinlei Zhou, Ran Wang, Chee Peng Lim, Xi-Zhao Wang, and QM Jonathan Wu. 2022. A review of generalized zero-shot learning methods. *IEEE transactions on pattern analysis and machine intelligence*, 45(4):4051–4070.

Somin Wadhwa, Silvio Amir, and Byron C Wallace. 2023. Revisiting relation extraction in the era of large language models. In *Proceedings of the conference. Association for Computational Linguistics. Meeting*, volume 2023, page 15566. NIH Public Access.

Jue Wang and Wei Lu. 2020. Two are better than one: Joint entity and relation extraction with table-sequence encoders. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1706–1721.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Xiang Wei, Xingyu Cui, Ning Cheng, Xiaobin Wang, Xin Zhang, Shen Huang, Pengjun Xie, Jinan Xu, Yufeng Chen, Meishan Zhang, et al. 2023. Zero-shot information extraction via chatting with chatgpt. *arXiv preprint arXiv:2302.10205*.

Ting Xu, Haiqin Yang, Fei Zhao, Zhen Wu, and Xinyu Dai. 2024. A two-agent game for zero-shot relation triplet extraction.

Luyao Zhu, Wei Li, Rui Mao, Vlad Pandelea, and Erik Cambria. 2023. Paed: Zero-shot persona attribute extraction in dialogues. In *Proceedings of the 61st Annual Meeting of the Association for Computational*

Linguistics (Volume 1: Long Papers), pages 9771–9787.

A Appendix

Table 6: The Prompt in ICL,ChatIE and RelationPrompt

Stage	ICL	ChatIE	RelationPrompt(GPT3.5) TAG+RelationPrompt(GPT3.5)
Stage-One	The given sentence is César Gaviria Trujillo Airport is an airport serving the town of Inírida in the Guainía Department of Colombia. List of given relations: [relation list] What relations in the given list might be included in this given sentence? If not present, answer: none. Respond in the form of (head entity1, tail entity1, relation1), (head entity2, tail entity2, relation2)	The given sentence is César Gaviria Trujillo Airport is an airport serving the town of Inírida in the Guainía Department of Colombia. List of given relations: [relaion list] What relations in the given list might be included in this given sentence? If not present, answer: none. Respond as a tuple, e.g. (relation 1, relation 2, ...)	Prompt for generating triplets from a relation. Given a relation, generate the head and tail entities to compose the relation triplet of the form (head entity, tail entity, relation). For example: Given the relation composer, we have triplets (Wolfgang Amadeus Mozart, Symphony No. 40., composer) Now given the relation: composer, please generate several triplets
Stage-Two	None	According to the given sentence, the relation between them is contains administrative territorial entity, find the head and tail entities and list them all by group if there are groups. If not present, answer: none. Respond in the form of (head entity1, tail entity1), (head entity2, tail entity2), ...	Prompt for generating sentences from a triplet. Generate a sentence with the given (head entity, tail entity, relation) triplet. For example: Given the triplet (Ludwig van Beethoven, Symphony No. 5., composer), we have sentence: Ludwig van Beethoven is the composer of Symphony No. 5. Now given the triplet: ('Ludwig van Beethoven', 'Symphony No. 9', 'composer'). Now given the triplet: ('Wolfgang Amadeus Mozart','Symphony No. 41', 'composer'). please generate the sentence

Table 7: Different Two-Stage Prompt

Stage	prompt1	prompt2
Stage-One	**Please solve the Relation selection task**.the context, Select the list of possible relationships present in the sentence.Relatioj type list is [type list].Context:[text]	**Please solve the Named Entity Extraction task**.Given the context, Extracting all the entities in this sentence.Context:{text}
Stage-Two	Please solve the Relation Extraction task. Combining the selected relation type list, provide at least one relation triplet in this sentence.The output format must be (head entity1, tail entity1), (head entity2, tail entity2). Do not have any other output except for relation triples;Only one triplet output is allowed)	Engage in the process of identifying connections between entities within a given sentence. Upon the extraction of the respective entities, it is mandatory to formulate at least one triplet that encapsulates the relationship between them. The types of relationships to be considered are confined to the ones provided in the aforementioned list. The format for the output should strictly adhere to the structure of (head entity1, relation type ,tail entity1)(head entity2, relation type,tail entity2). Any deviation from this format or inclusion of additional content outside of the specified triplets is not permissible.