Generation-Augmented Retrieval: Rethinking the Role of Large Language Models in Zero-Shot Relation Extraction

Zehan Li, Fu Zhang*, Tianyue Peng, He Liu, Jingwei Cheng
School of Computer Science and Engineering, Northeastern University, China
Key Laboratory of Intelligent Computing in Medical Image of Ministry of Education
Northeastern University, China
{lizehan1999,pengtianyue0716,liuhe0607}@163.com,
{zhangfu,chengjingwei}@neu.edu.cn

Abstract

Recent advances in Relation Extraction (RE) emphasize Zero-Shot methodologies, aiming to recognize unseen relations between entities with no annotated data. Although Large Language Models (LLMs) have demonstrated outstanding performance in many NLP tasks, their performance in Zero-Shot RE (ZSRE) without entity type constraints still lags behind Small Language Models (SLMs). LLM-based ZSRE often involves manual interventions and significant computational overhead, especially when scaling to large-scale multi-choice data.

To this end, we introduce **RE-GAR-AD**, which not only leverages the generative capability of LLMs but also utilizes their representational power without tuning LLMs. We redefine LLM-based ZS**RE** as a retrieval challenge, utilizing a **G**eneration-**A**ugmented **R**etrieval framework coupled with a retrieval **AD**juster. Specifically, our approach guides LLMs through crafted prompts to distill sentence semantics and enrich relation labels. We encode sentences and relation labels using LLMs and match their embeddings in a triplet fashion. This retrieval technique significantly reduces token input requirements. Additionally, to further optimize embeddings, we propose a plug-in retrieval adjuster with only 2M parameters, which allows rapid fine-tuning without accessing LLMs' parameters. Our LLM-based model demonstrates comparable performance on multiple benchmarks.[1]

1 Introduction

Relation Extraction (RE) aims to select the relation from a predefined set for two target entities in a given sentence. Traditional supervised RE methods (Wu and He, 2019; Zheng et al., 2021) demand significant annotated data and struggle to handle unseen relations effectively. Recently, there has been a growing interest in Few-Shot RE (FSRE) (Yang et al., 2020) and Zero-Shot RE (ZSRE) (Levy et al., 2017; Gong and Eldardiry, 2021), which have limited or no labeled data for novel relations.

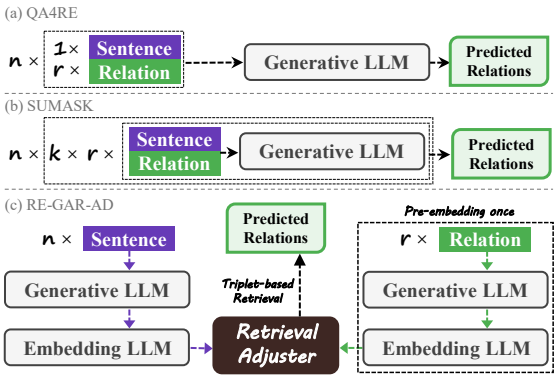

Figure 1: Comparison between (a) QA4RE (Zhang et al., 2023), (b) SUMASK (Li et al., 2023), and (c) our approach.

Existing research on ZSRE typically involves fine-tuning Small Language Models (SLMs) [2] on predicting unseen relations (Zhao et al., 2023a; Li et al., 2024c). Large Language Models (LLMs) have demonstrated impressive performance in various downstream natural language processing (NLP) tasks (Zhao et al., 2023b), leveraging techniques such as In-Context Learning (ICL) (Ma et al., 2023c) and Chain of Thought (CoT) (Ma et al., 2023a) to address FSRE. Recently, LLM-based ZSRE studies have also made significant progress. QA4RE (Zhang et al., 2023) redefines ZSRE as a Question Answering (QA) task and manually construct templates for relation labels. It requires token inputs at the $n \times r$ level for n input sentences and r candidate relations (Figure 1 (a)). However, the inherent selection bias of LLMs may result in poor robustness (Zheng et al., 2024). SUMASK (Li et al., 2023) determines relations by calculating

[1] Code is available at https://github.com/lizehan1999/RE-GAR-AD.

* Corresponding author.

[2] We define SLMs as language models with less than 1B parameters that can be easily fine-tuned (Ma et al., 2023c).

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 17928–17941
November 4-9, 2025

uncertainty across k responses of LLMs. The recursive input leads to a significant $n \times k \times r$ token input overhead (Figure 1 (b)), restricting its scalability on large-scale data. Therefore, making ZSRE more efficient and accurate while benefiting from LLMs is a key concern for the current system.

In this work, we explore new possibilities for LLM-based ZS**RE** and propose the **RE-GAR-AD** framework, which is based on the Generation-Augmented Retrieval (**GAR**) paradigm and incorporates a retrieval **AD**juster. Considering the potential of LLMs as encoders for obtaining embeddings (BehnamGhader et al., 2024), we rethink and redefine LLM-based ZSRE as an embedding-based retrieval problem. We input generation-augmented sentences and relation labels separately to LLMs to obtain their embeddings and conduct sentence and relation retrieval through matching scores. All candidate relation labels only need to be input once and kept fixed, requiring only token inputs at $n + r$ level as shown in Figure 1 (c). This not only reduces the inference cost of large-scale multi-choice ZSRE data for LLMs but also demonstrates performance comparable to that of existing methods.

Specifically, we leverage the generation capability of LLMs to improve retrieval efficiency. Considering that a sentence may contain more than one relation pattern, we distill valuable information for RE from *sentences* using Sentence Distillation Prompt, which includes the types of two target entities and their possible semantic relation. Additionally, we guide LLMs to utilize their internal general knowledge to generate deep information about *relation labels* through the Relation Enrichment Prompt, which explores possible entity types relevant to the relation. We then use LLMs to encode and obtain embeddings of sentences and relation labels, and achieve more accurate relation retrieval through Triplet-based Retrieval. Furthermore, SLMs-based ZSRE fine-tuning typically involves learning prior knowledge from seen data, which may be computationally expensive for LLMs and impractical for API-based LLMs (Yoon et al., 2023; Li et al., 2024b). Therefore, we carefully design a plug-in Retrieval Adjuster for the framework. It aims to learn the ability to align sentence and relation label representations from seen data, and further adjust and optimize the embeddings of unseen data without accessing the full parameters of LLMs, thereby improving retrieval efficiency. Our contributions are summarized as follows:

- We redefine LLM-based ZSRE as a retrieval task and propose a novel Generation-Augmented Retrieval (GAR) framework, significantly reducing the token inputs while ensuring accuracy.
- We carefully design two prompts further to process the semantics of sentences and relation labels using the generative ability of LLMs, improving retrieval efficiency.
- We introduce a retrieval adjuster plugin for the GAR framework, with only 2M parameters, which can enhance retrieval performance without accessing the parameters of LLMs.
- Experimental results show that RE-GAR-AD achieves competitive performance across five ZSRE benchmark datasets. Without involving fine-tuning, our approach outperforms the state-of-the-art LLM-based method by **6.93**.

2 Related Work

2.1 LLMs for Language Representation

Large Language Models (LLMs) are typically trained on large corpora and have demonstrated advanced performance across many downstream NLP tasks (Zhao et al., 2023b). For tasks such as retrieval and clustering that require sentence embeddings, a common approach involves leveraging the representation of the end-of-sequence token of LLMs as sentence embeddings (Ma et al., 2023b). Considering that causal attention might limit the representation capability of decoder-only LLMs, recent works introduce bidirectional attention (Muennighoff et al., 2024) and echo mechanisms (Springer et al., 2024) to overcome limitations. These works demonstrate that generative LLMs are also proficient at language representation (Wang et al., 2023), and any robust decoder-only LLMs can produce universal text embeddings with minor adaptation (BehnamGhader et al., 2024).

2.2 LLMs for Relation Extraction

The existing work on LLM-based RE can be categorized into three paradigms: **Tuning-based methods** involve fine-tuning LLMs with accessible parameters (Kim et al., 2023; Sun et al., 2024). Recent works adapt LLMs to RE tasks through annotation guidelines (Sainz et al., 2024) or tabular prompting (Li et al., 2024a). **ICL-based methods** facilitate In-Context Learning (ICL) (Min

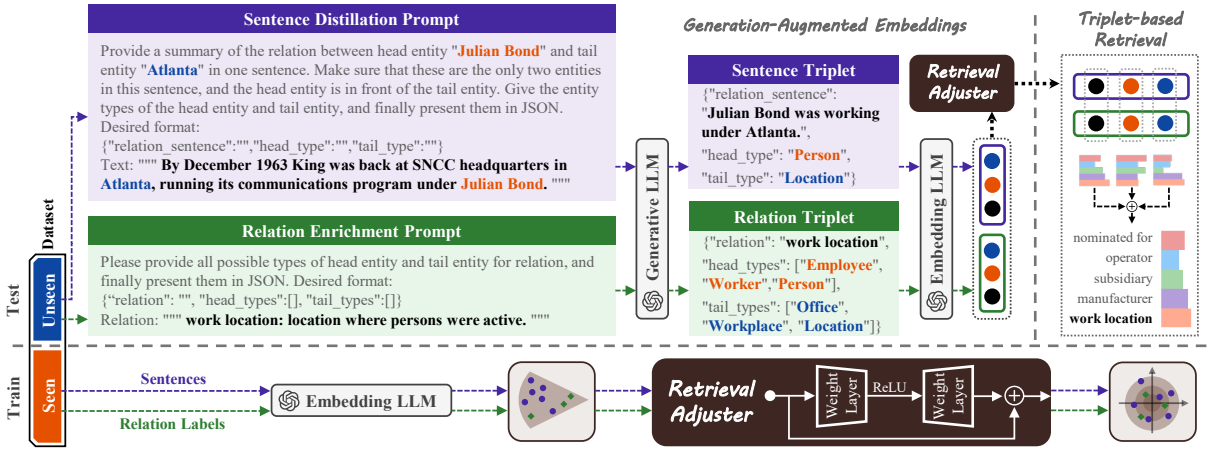

Figure 2: An overview of the proposed RE-GAR-AD framework. It consists of three key components: (1) **Generation-Augmented Embeddings**, where Generative LLMs distill semantic information from sentences and relation descriptions using two prompts; (2) **Retrieval Adjuster**, which calibrates embeddings using relation patterns learned from seen data; and (3) **Triplet-based Retrieval**, where embedding triplets are matched to infer the final relation label.

et al., 2022) by observing demonstrations within prompts without updating the LLMs parameters, which is applied in FSRE. They use the Retrieval-Augmented Generation (RAG) paradigm (Rubin et al., 2022) to retrieve suitable demonstrations, which may include quality analysis (Wan et al., 2023) or logical reasoning (Ma et al., 2023c). **Instruction-based methods** execute tasks through explicit instructions such as multiple-choice QA (Zhang et al., 2023) or uncertainty estimation (Li et al., 2023) without demonstrations, making them applicable to ZSRE.

A common limitation of the above methods is the increase in candidate relations along with a large number of token inputs, which consumes time and resources in large-scale ZSRE. Additionally, post-processing may be required to address output format issues. To address these challenges, we propose a novel LLM-based paradigm called Generative-Augmented Retrieval (GAR). Unlike RAG, GAR directly identifies relations through retrieval, fundamentally avoiding format issues and reducing computational costs by separately modeling sentences and relations. Furthermore, we devise a retrieval adjuster that can be plugged into LLMs to further improve ZSRE performance.

3 Methodology

3.1 Problem Formulation

Given a predefined set of seen relations R_s and unseen relations R_u, with no overlap between them. Given seen data D_s and unseen data D_u. The ZSRE's objective is to predict $r_u \in R_u$ between the head entity e^1 and the tail entity e^2 in the input sentence x_u for unseen instance $I_u = (x_u, e^1_u, e^2_u) \in D_u$. Before prediction, learning can be conducted from seen instances $I_s = (x_s, e^1_s, e^2_s, r_s) \in D_s, r_s \in R_s$, and the knowledge can be transferred to predict.

3.2 Overview of RE-GAR-AD

The proposed RE-GAR-AD framework, which redefines LLM-based ZSRE as an embedding-based retrieval problem, consists of three modules as shown in Figure 2:

(1) **Generation-Augmented Embeddings** involves the use of Generative LLMs to distill and enrich sentences and relation labels through two carefully designed prompts. These are then encoded by Embedding LLMs to obtain fixed-dimensional embedding triplets. (2) **Retrieval Adjuster** learns from the relation patterns of the seen dataset D_s and is used to adjust the embedding triplets corresponding to the unseen sentences and relations, thereby improving retrieval performance. It is worth noting that we use the same data processing method in the testing phase as in the training phase, applying both Generation Augmentation and Retrieval Adjustment. (3) Through **Triplet-based Retrieval**, the adjusted sentence embedding triplets are matched with the adjusted relation embedding triplets, and the highest scoring relation is chosen as the final relation.

3.3 Generation-Augmented Embeddings

Sentence Distillation Prompt. To distill crucial content for RE from sentences with redundant infor-

mation, we propose a Sentence Distillation Prompt comprising three carefully crafted instructions. For a given input sentence x_u with two target entities e^1, e^2, the *Relation Summarization* instruction requires LLMs to encapsulate the semantic relation between the annotated entities within one sentence, denoted as x_u^*. This requires LLMs to utilize their summarization abilities. The *Type Identification* instruction aims to infer the types of the head and tail entities that are crucial for RE (Lyu and Chen, 2021), represented as T_u^1 and T_u^2, respectively. The final triplet output is organized into a JSON object using the *Triplet Combination* instruction, described as $\{x_u^*, T_u^1, T_u^2\}$.

Relation Enrichment Prompt. To leverage the general knowledge of LLMs for mining deep information from relation labels and descriptions, we introduce a Relation Enrichment Prompt. It aims to instruct LLMs to freely generate all possible entity types $\hat{T}_u$ associated with relation labels. This automated process effectively alleviates the manual labor involved in annotating head entity types and tail entity types for relations. Subsequently, we organize them into triplets in JSON format: $\left\{r_u, \{\hat{T}_{u,i}^1\}_{i=1}^n, \{\hat{T}_{u,j}^2\}_{j=1}^m\right\}$.

Embedding with LLMs. To match sentence triplets and relation triplets, we embed them into the same embedding space. We obtain an embedding $h \in \mathbb{R}^d$ for each element of the sentence triplets and relation triplets by calling the open API of the latest GPT series model "*text-embedding-3-small*", which is trained via matryoshka representation learning (Kusupati et al., 2022). It is noteworthy that for $\hat{T}_u$, we conduct average pooling for each set to obtain $h_{\hat{T}_u}$ of entity types. Each sentence embedding triplet and relation embedding triplet can be represented respectively as $\text{Tri}_x = \{h_{x_u^*}, h_{T_u^1}, h_{T_u^2}\}$ and $\text{Tri}_r = \{h_{r_u}, h_{\hat{T}_u^1}, h_{\hat{T}_u^2}\}$.

3.4 Retrieval Adjuster

Embeddings from LLMs have a potential *anisotropy problem* due to common-word biases (Li and Li, 2024), causing the embeddings to occupy a narrow cone. Therefore, contrastive learning-based fine-tuning is required to distribute the embeddings uniformly (Gao et al., 2021).

However, fine-tuning LLMs for specific tasks is resource-intensive and necessitates full access to the parameters of the LLMs, posing challenges with API-based LLMs. Therefore, we propose an independent *retrieval adjuster* (AD) for adapting ZSRE, aimed at learning the ability to align sentence and relation label representations from seen data and further adjust the unseen embeddings. AD does not require access to LLMs' parameters and can easily plug into LLMs.

To adapt to the task while retaining the LLM-based embedding semantics, AD integrates residual features learned by network $\text{F}(\cdot)$ with the original embeddings through skip connections to produce the final embedding $\tilde{h}$:

$$\tilde{h} = h + \text{F}(h). \tag{1}$$

We employ a straightforward fully connected network as $\text{F}(\cdot)$, with detailed operations as follows:

$$\text{F}(h) = W_2(\text{ReLU}(W_1(h) + b_1)) + b_2, \tag{2}$$

where $W_1 \in \mathbb{R}^{d\times n}$, $W_2 \in \mathbb{R}^{n\times d}$ are parameter matrices, and b_1, b_2 are bias. Notably, inspired by Yoon et al. (2023), we employ a shared $\text{F}(\cdot)$ for all embeddings, thereby maintaining a unified embedding space to achieve better retrieval.

3.5 Triplet-based Retrieval

To enable Tri_x to retrieve the corresponding Tri_r, we introduce triplet score c to compute the similarity coefficient:

$$\begin{aligned} c(\text{Tri}_x, \text{Tri}_r) = \alpha\langle h_{x^*}, h_r\rangle + \\ (1-\alpha)\left(\langle h_{T^1}, h_{\hat{T}^1}\rangle + \langle h_{T^2}, h_{\hat{T}^2}\rangle\right), \end{aligned} \tag{3}$$

where α is hyperparameter, and $\langle , \rangle$ denotes the cosine similarity. We train AD on seen data D_s to align the embeddings of Tri_{x_s} and Tri_{r_s}. The training objective of $\text{F}(\cdot)$ is to maximize the triplet score between $\tilde{\text{Tri}}_{x_s}^i$ and positive $\tilde{\text{Tri}}_{r_s}^i$, while minimizing its score with negative $\tilde{\text{Tri}}_{r_s}^j$:

$$\begin{aligned} \mathcal{L} = \frac{1}{N}\sum_{i=1}^{N} \max\{0, \max(c(\tilde{\text{Tri}}_{x_s}^i, \tilde{\text{Tri}}_{r_s}^j)) \\ - c(\tilde{\text{Tri}}_{x_s}^i, \tilde{\text{Tri}}_{r_s}^i) + \Delta\}, i \neq j, \end{aligned} \tag{4}$$

where N denotes the batch size, and Δ represents a predefined threshold. Negative samples are randomly selected within the same batch, maintaining a 1:3 ratio of positive to negative samples. This approach can promote the uniform distribution of sentence embeddings and relation embeddings, thereby alleviating the *anisotropy problem*.

The trained AD is utilized to adjust the embeddings of unseen Tri_{x_u} and Tri_{r_u} during the testing phase. The highest computed $c(\tilde{\text{Tri}}_{x_u}, \tilde{\text{Tri}}_{r_u})$ is chosen as the final relation.

4 Experiments Setup

4.1 Datasets

We conduct experiments on five benchmarks. Following established setups (Zhao et al., 2023a), we conduct our primary experiments on two widely used datasets: **Wiki-ZSL** (Chen and Li, 2021) and **FewRel** (Han et al., 2018). Wiki-ZSL contains 94,383 instances and 113 relations, and is acquired through distant supervision from Wiki-KB, characterized by imbalanced samples and heightened noise. FewRel contains 56,000 instances and 80 relations, and is meticulously annotated, ensuring equal samples per relation. Consistent with prior works, we divide the datasets into a training set containing seen relations and a validation set containing unseen relations, focusing on evaluating under unseen relations $m = 5, 10, 15$. To ensure the reliability of results, we perform 10-fold cross-validation and report the average results. Notably, the entity types are not provided in Wiki-ZSL and FewRel, posing challenges to ZSRE as candidate relations cannot be filtered based on entity types.

Moreover, following the setups of other LLM-based methods (Li et al., 2023), we also select three datasets where entity types are given: (1) **TACRED** (Zhang et al., 2017) contains 42 relations and 17 entity types, (2) **TACREV** (Alt et al., 2020) contains 42 relations and 17 entity types, and (3) **Re-TACRED** (Stoica et al., 2021) contains 40 relations and 17 entity types. We further evaluate on their test sets with the none-of-the-above (NoTA) relation excluded.

4.2 Implementation Details

We set `temperature = 0` for GPT series models to improve the reproducibility. We apply the same prompt for Wiki-ZSL and FewRel. Since the entity types are seen in TACRED, TACREV, and Re-TACRED, we remove instructions for generating entity types from prompt. Following previous work, we filter a subset of candidate relations based on the entity types. For *retrieval adjuster* AD, we set the same hyperparameters for all experiments. We use AdamW with a learning rate of $1e-5$, 10 epochs, and a batch size of 64. We set the dimension of the hidden layer n to $d/2$, resulting in approximately 2.36M parameters for AD, which is much smaller than *BERT-based* SLMs. We conducted a grid search on FewRel and determined hyperparameters $\alpha = 0.75, \Delta = 0.1$, negative ratio$= 1:3$ when the model performed the best.

4.3 Baseline Models

SLMs-based methods. We compare: *embedding-based methods* R-BERT (Wu and He, 2019), ZS-BERT (Chen and Li, 2021), and RE-Matching (Zhao et al., 2023a); *text entailment-based* methods ESIM (Chen et al., 2017), LaVeEntail (Sainz et al., 2021); and *generation-based methods* RE-Prompt (Chia et al., 2022), SuRE (Lu et al., 2022).
LLM-based methods. We compare previous competitive baselines, including VanillaPrompt, QA4RE, SUMASK, and MICRE. VanillaPrompt and QA4RE are from Zhang et al. (2023), respectively guide LLMs to generate relation labels or multiple-choice options, where QA4RE involves manually building relation templates. As they did not evaluate on Wiki-ZSL and FewRel, we rerun the results based on the original paper and available code. SUMASK (Li et al., 2023) classifies relations by generating multiple responses for all candidate relations and estimating uncertainty from LLMs. MICRE (Li et al., 2024a) uses 12 public datasets to fine-tune LLaMA (Touvron et al., 2023a) through tabular prompting, achieving performance competitive with SLMs. Following the previous approach, we randomly sample 1,000 examples as test data from each dataset to control the cost of comparison with the LLM-based baselines. Since using seen data for training the AD incurs additional costs and may not be available in some datasets, we provide two sets of results indicating whether seen data are used to train AD, denoted as **RE-GAR** and **RE-GAR-AD**, respectively.

5 Analysis and Discussion

5.1 Main Result

We report the main results on Wiki-ZSL and FewRel in Table 1. Overall, our approach demonstrates competitive performance across all settings on both datasets.

(1) Compared to LLM-based methods, our approach ensures strong performance even without considering seen data (RE-GAR's average F1 is **6.93** higher than QA4RE across six settings). We attribute this to the novel GAR framework, which effectively utilizes both the generative and representational capabilities of LLMs, allowing LLMs to unleash more potential on ZSRE tasks. Additionally, we observe that when considering seen data, RE-GAR-AD achieves even stronger performance with fewer training samples (with an F1 advantage of **3.71** over MICRE).

Methods	Wiki-ZSL									FewRel								
	m=5			m=10			m=15			m=5			m=10			m=15		
	P.	R.	F1	P.	R.	F1	P.	R.	F1	P.	R.	F1	P.	R.	F1	P.	R.	F1
– SLMs-based																		
R-BERT*	39.22	43.27	41.15	26.18	29.69	27.82	17.31	18.82	18.03	42.19	48.61	45.17	25.52	33.02	25.52	16.95	19.37	18.08
ESIM*	48.58	47.74	48.16	44.12	45.46	44.78	27.31	29.62	28.42	56.27	58.44	57.33	42.89	44.17	43.52	29.15	31.59	30.32
ZS-BERT*	71.54	72.39	71.96	60.51	60.98	60.74	34.12	34.38	34.25	76.96	78.86	77.90	56.92	57.59	57.25	35.54	38.19	36.82
LaVeEntail*	77.39	75.90	76.63	71.86	71.14	71.50	62.13	61.76	61.95	91.14	90.86	91.00	83.05	82.55	82.80	72.83	72.10	72.46
REPrompt*	70.66	83.75	76.63	68.51	74.76	71.50	63.69	67.93	65.74	90.15	88.50	89.30	80.33	79.62	79.96	74.33	72.51	73.40
RE-Matching*	78.19	78.41	78.30	74.39	73.54	73.96	67.31	67.33	67.32	92.82	**92.34**	92.58	83.21	**82.64**	**82.93**	73.80	73.52	73.66
– LLM-based																		
VanillaPrompt	79.83	56.56	66.21	62.98	58.05	60.41	58.54	51.26	54.66	74.87	51.15	60.77	65.28	46.46	54.29	59.70	50.56	54.76
QA4RE	87.20	74.97	80.62	65.80	58.25	61.80	62.17	57.80	59.90	88.00	75.66	81.37	79.00	73.26	76.02	65.33	67.45	66.37
SUMASK†	75.64	70.96	73.23	62.31	61.08	61.69	43.55	40.27	41.85	78.27	72.55	75.30	64.77	60.94	62.80	44.76	41.13	42.87
MICRE†	76.46	78.53	77.48	72.36	74.88	73.60	67.14	68.87	67.99	89.34	91.88	90.59	80.67	82.31	81.48	73.74	**75.83**	74.77
– Ours (LLM-based)																		
RE-GAR	85.13	83.85	84.44	74.98	73.32	74.12	68.45	66.35	67.37	90.75	89.40	90.07	80.89	78.76	79.81	73.48	70.26	71.82
RE-GAR-AD	**88.96**	**88.42**	**88.63**	**77.63**	**77.85**	**77.70**	**70.80**	**69.26**	**70.02**	**93.03**	92.16	**92.59**	**83.48**	82.10	82.78	**77.69**	75.31	**76.47**

Table 1: Experimental results on Wiki-ZSL and FewRel. We mark the best results in **bold** and the second-best underlined. All generation based on API are run on *GPT-3.5-turbo*. **RE-GAR** and **RE-GAR-AD** respectively represent the results without using seen data, and the results of fine-tuning our retrieval adapter with only seen data. The results indicated by † and ∗ are reported in MICRE (Li et al., 2024a) and RE-Matching (Zhao et al., 2023a). P., R., and F1 denote Precision, Recall, and F1 score, respectively.

(2) Compared to SLMs-based methods trained on seen data, RE-GAR-AD achieves up to **5.59** higher F1 on Wiki-ZSL than the existing SOTA model, RE-Matching. Interestingly, our method outperforms other baselines by a large margin at $m = 5$ on Wiki-ZSL, and we attribute this improvement partly to the effective noise reduction in the distant supervision dataset through *Sentence Distillation*. It is worth noting that the parameter of the fine-tuned retrieval adjuster is much smaller than BERT, which is adopted as the backbone for most SLM-based methods.

(3) Additionally, in Table 2, we report the micro-averaged F1 on TACRED, TACREV, and Re-TACRED. Since entity types are provided, we follow the previous works (Zhang et al., 2023; Li et al., 2023) by discarding irrelevant relations. The results indicate that RE-GAR achieves comparable performance on three datasets.

5.2 Ablation Experiment

We report the impact of different components on RE-GAR-AD, as shown in Table 3.

1. We use Embedding LLMs to encode the raw sentences when we remove the *Sentence Distillation*. The average performance drops by 7.74. This indicates that the summarization capability of LLMs can effectively reduce irrelevant information in sentences, allowing them to focus more on the semantic association between entities, thus gaining an advantage in relation matching.

2. We manually annotate possible entity types for relations when we remove the *Relation Enrichment*. We observe a slight decrease in average performance by 1.67, indicating that LLMs have the potential to generate data of comparable quality to manually annotated data, and that richer information may yield greater benefits.

3. We solely match sentences and relation labels

Methods	TACRED	TACREV	Re-TACRED
LaVeEntail$_{\text{BART}}$‡	51.4	55.3	44.0
LaVeEntail$_{\text{DeBERTa}}$‡	55.1	57.2	**64.3**
SuRE$_{\text{BART}}$‡	20.4	22.2	23.6
SuRE$_{\text{PEGASUS}}$‡	21.8	21.6	22.4
VanillaPrompt‡	48.1	51.0	55.3
QA4RE‡	59.4	59.4	61.2
SUMASK†	57.7	-	-
RE-GAR	**64.8**	**63.4**	53.8

Table 2: Experimental results on TACRED, TACREV, and Re-TACRED. The results indicated by † and ‡ are reported in SUMASK (Li et al., 2023) and QA4RE (Zhang et al., 2023).

Methods	m=5	m=10	m=15
RE-GAR-AD	**92.59**	**82.78**	**76.47**
w/o. Sentence Distillation	88.45	73.06	67.12
w/o. Relation Enrichment	92.06	80.13	74.64
w/o. Triplet-based Retrieval	82.18	68.44	60.32
w/o. Retrieval Adjuster	90.07	79.81	71.82

Table 3: Ablation study of RE-GAR-AD. We report the F1 performance on FewRel under $m = 5, 10, 15$.

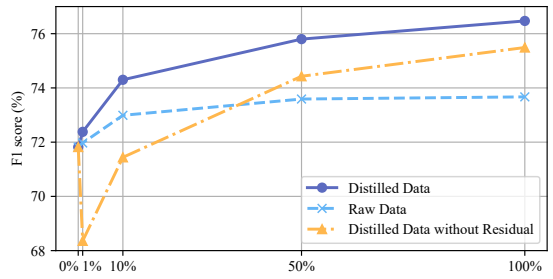

Figure 3: Effect of data scale, data quality, and residual connections on *retrieval adjuster* AD.

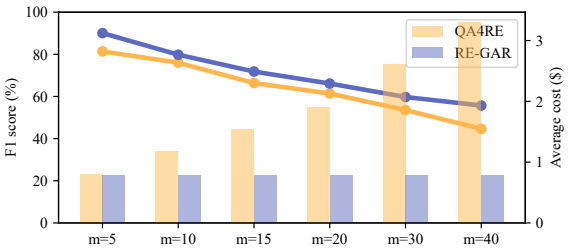

Figure 4: Comparison of API overhead (Bar) and F1 score (Line) for 8,000 instances. To control the cost, we randomly sample 100 samples for each relation.

when we remove the *Triplet-based Retrieval* (i.e., without considering entity type information in the retrieval), there is a dramatic decline in average performance (-13.63), which proves the crucial role of entity information in relation retrieval.

4. When *Retrieval Adjuster* AD is removed, it means that we do not benefit from fine-tuning, leading to an average performance drop of 3.42.

5.3 Analysis on Retrieval Adjuster

Retrieval adjuster AD needs to be trained on seen data, which may introduce additional generation costs and training overhead. Therefore, we further investigate the impact of training data of varying scales and qualities on the performance of AD under the setting of $m = 15$ on the FewRel dataset. Specifically, we randomly extract $\{1\%, 10\%, 50\%\}$ of the samples from the seen data to construct subsets, maintaining an even balance of samples per relation.

Data scale. As shown in Figure 3, more training samples can bring greater benefits to AD, which is consistent with intuition. Notably, only 10% of the distilled seen data can lead to a 2.48 F1 score increase, while increasing the data scale tenfold only further improves the F1 score by 2.17.

Data quality. Even if the data is not distilled (raw data), AD can still enhance performance without negatively impacting the model. Since this strategy does not require LLMs to generate, it can be more cost-effective in some scenarios.

Residual connection. We observe that removing residual connection when the training data scale is small leads to a performance decrease, indicating that residual connection can effectively improve information flow and reduce overfitting. However, with sufficient data scale, the additional benefits of residual connection may not be as apparent.

5.4 Model Efficiency

To evaluate our approach from both performance and cost perspectives, we compare it with LLM-based baselines on FewRel with settings of $m = 5, 10, 15, 20, 30, 40$ and report the results in Figure 4. For a fair comparison, we only report the performance of RE-GAR, which does not benefit from fine-tuning on seen data. Moreover, as described in Section 1, SUMASK requires an input overhead of $n \times k \times r$ tokens. Therefore, we choose QA4RE as the baseline, which has a more efficient input overhead $n \times r$. The results are derived from *GPT-3.5-turbo* and *text-embedding-3-small*. The estimation of API costs is sourced from OpenAI [3]

The results indicate that RE-GAR consistently demonstrates advantages in both F1 score and cost compared to QA4RE across all settings of m. Moreover, as m increases, RE-GAR's performance is only mildly affected. Additionally, since relation labels serving as retrieval targets are only pre-embedded once and fixed, the average cost of RE-GAR in large-scale ZSRE scenarios only undergoes slight changes as m increases.

5.5 Scalability Analysis

We conduct scalability research on SLMs, open-source LLMs, and API-based LLMs, and report F1 score on FewRel with $m = 15$. We select 100 samples for each relation to build subsets to control costs. Note that the *generation-augmentation* and *retrieval* are executed in a pipelined manner, and their combination is not fixed.

In Figure 5, we compare different embedding models. For SLMs, we choose *SRoBERTa* (Reimers and Gurevych, 2019), *PromptRoBERTa* (Jiang et al., 2022), *SimCSE-Roberta* (Gao et al., 2021), and *BGE-en* (Xiao et al., 2023). For open-source LLMs, we introduce LLM2Vec (BehnamGhader et al., 2024), which converts *LLaMA2* (Touvron et al., 2023b) and *Mistral* (Jiang et al., 2023) into general text encoders. In addition,

[3] https://openai.com/api/pricing/

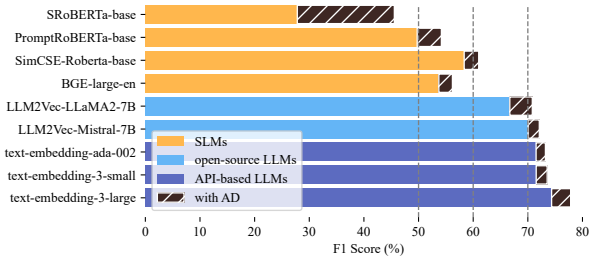

Figure 5: Scalability analysis: Generate with GPT-3.5-turbo. Embedding with different SLMs and LLMs.

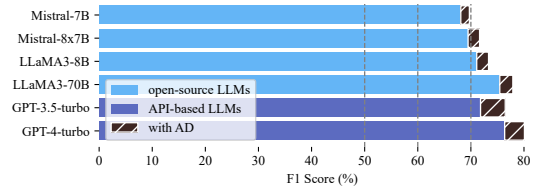

Figure 6: Scalability analysis: Embedding with *text-embedding-3-small*. Generate with different LLMs.

we explore different API-based LLMs (OpenAI *text-embedding* (Neelakantan et al., 2022)). We observe that the performance of SLMs generally does not exceed 61, which may indicate a significant gap in representation capabilities between SLMs and LLMs on ZSRE, revealing the potential of LLMs in representation. RE-GAR-AD demonstrates excellent performance on all LLMs, proving its robustness.

In Figure 6, we compare different generative models, including *LLaMA3*, *Mistral*, *GPT-3.5-turbo*, and *GPT-4-turbo*, which achieve remarkable performance with their strong generation capabilities. We observe that more powerful LLMs such as *text-embedding-3-large* and *GPT-4-turbo* can provide sustained benefits in RE-GAR-AD.

Furthermore, we notice that our proposed plug-in retrieval adjuster consistently shows improvement, indicating that it is model-agnostic.

5.6 Error Analysis and Case Study

To investigate the effectiveness of our RE-GAR framework, we provide a typical example as shown in **Case 1** of Figure 7. For QA4RE, we construct relation-entailed templates as label verbalization: "*P27: country of citizenship*" is processed into "*[Andrew Goldberg] is a citizen of [United States]*", and "*P361: part of*" is processed into "*[Andrew Goldberg] is part of [United States]*". Due to the semantic similarity between the two options, generative LLMs might confuse them and output the incorrect option. RE-GAR refines the semantic relationship between *Andrew Goldberg* and the *United States*, yielding the accurate information: "*Andrew Goldberg is a writer from the United States.*" Additionally, it pays extra attention to the type information of the two entities (*Person* and *Country*).

Sentence	After leaving Island Records due to a disagreement over the band's artistic direction, Fightstar signed with the independent label Institute Records for their second album.	
Head Entity: Fightstar		**Tail Entity:** Island Records
Distilled Sentence	Fightstar left Island Records due to a disagreement over artistic direction. Band. Record Label.	
Enriched relation (P175)	Actor, Musician, Band	Musical work
Enriched relation (P264)	Music recordings, Music videos	Record label, Brand, Trademark
Ground Truth	P264: record label	✓
QA4RE	P175: performer	✗
RE-GAR	P175: performer	✗
RE-GAR-AD	P264: record label	✓

Figure 7: Case study on FewRel with $m = 15$. **Sentence**, **Head Entity**, and **Tail Entity** are derived from the raw data. **Distilled Sentence** represents the data distilled through the RE-GAR-AD framework.

As shown in **Case 2** of Figure 7, when LLMs fail to capture meaningful semantic connections between *Fightstar* and *Island records*, it can lead to ineffective retrieval. Our proposed plug-in retrieval adjuster addresses this by learning relation patterns from seen data and subsequently adjusting when retrieving unseen relation labels, successfully categorizing the relation as "*record label*".

Additionally, we report the performance of our framework on 80 relation categories in Appendix C. We observe that although RE-GAR-AD can distinguish similar relations better with the retrieval adjuster, challenges still exist when co-occurring relations are present in sentences. This will be one of our future optimization directions.

6 Conclusion

We present RE-GAR-AD, a simple and effective ZSRE framework based on LLMs. We introduce a novel LLM-based ZSRE paradigm of Generation-Augmented Retrieval, leveraging LLMs' generative and representational capabilities. Additionally, we train a plug-in retrieval adjuster to further optimize embeddings. Extensively evaluated on common ZSRE benchmarks, RE-GAR-AD demonstrates comparable performance across various settings, especially in cases where entity types are excluded. Furthermore, our method significantly improves the efficiency of LLMs in multi-choice relation extraction, which is advantageous for real-world ZSRE and opens up interesting avenues for future zero-shot classification efforts.

Limitations

Although the proposed framework demonstrates advanced performance and cost-effectiveness on large-scale ZSRE, its cost-effectiveness is not evident when the number of input sentences n is small. Additionally, due to API limitations, we are unable to obtain complete results from more powerful models such as GPT-4 and others. Furthermore, we employ only a straightforward structure for the retrieval adjuster, and we believe that further research on LLM-embedding-based adjusters would be an interesting direction. Finally, since our method encounters difficulties in handling the none-of-the-above (NoTA) relation, it may not be suitable for cases where the dataset contains the NoTA relation.

Acknowledgments

The authors sincerely thank the reviewers for their valuable comments, which improved the paper. The work is supported by the National Natural Science Foundation of China (62276057).

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*, pages 1558–1569.

Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. 2024. Llm2vec: Large language models are secretly powerful text encoders. *arXiv preprint arXiv:2404.05961*.

Chih Yao Chen and Cheng-Te Li. 2021. Zs-bert: Towards zero-shot relation extraction with attribute representation learning. In *NAACL 2021*, pages 3470–3479.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced lstm for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 1657–1668.

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.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *EMNLP 2021*, pages 6894–6910.

Jiaying Gong and Hoda Eldardiry. 2021. Zero-shot relation classification from side information. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 576–585.

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 *EMNLP 2018*, pages 4803–4809.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, and 1 others. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Ting Jiang, Jian Jiao, Shaohan Huang, Zihan Zhang, Deqing Wang, Fuzhen Zhuang, Furu Wei, Haizhen Huang, Denvy Deng, and Qi Zhang. 2022. Promptbert: Improving bert sentence embeddings with prompts. In *EMNLP 2022*, pages 8826–8837.

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.

Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham Kakade, Prateek Jain, and 1 others. 2022. Matryoshka representation learning. *Advances in Neural Information Processing Systems*, 35:30233–30249.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. In *CoNLL 2017*, pages 333–342.

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*, pages 6877–6892.

Guozheng Li, Peng Wang, Jiajun Liu, Yikai Guo, Ke Ji, Ziyu Shang, and Zijie Xu. 2024a. Meta in-context learning makes large language models better zero and few-shot relation extractors. *arXiv preprint arXiv:2404.17807*.

Haitao Li, Qingyao Ai, Jia Chen, Qian Dong, Zhijing Wu, Yiqun Liu, Chong Chen, and Qi Tian. 2024b. Blade: Enhancing black-box large language models with small domain-specific models. *arXiv preprint arXiv:2403.18365*.

Xianming Li and Jing Li. 2024. Bellm: Backward dependency enhanced large language model for sentence embeddings. In *NAACL 2024*, pages 792–804.

Zehan Li, Fu Zhang, and Jingwei Cheng. 2024c. Alignre: An encoding and semantic alignment approach for zero-shot relation extraction. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 2957–2966.

Keming Lu, I-Hung Hsu, Wenxuan Zhou, Mingyu Derek Ma, and Muhao Chen. 2022. Summarization as indirect supervision for relation extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6575–6594.

Shengfei Lyu and Huanhuan Chen. 2021. Relation classification with entity type restriction. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 390–395.

Xilai Ma, Jing Li, and Min Zhang. 2023a. Chain of thought with explicit evidence reasoning for few-shot relation extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2334–2352.

Xueguang Ma, Liang Wang, Nan Yang, Furu Wei, and Jimmy Lin. 2023b. Fine-tuning llama for multi-stage text retrieval. *arXiv preprint arXiv:2310.08319*.

Yubo Ma, Yixin Cao, Yong Hong, and Aixin Sun. 2023c. Large language model is not a good few-shot information extractor, but a good reranker for hard samples! In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10572–10601.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? In *EMNLP 2022*, pages 11048–11064.

Niklas Muennighoff, Hongjin Su, Liang Wang, Nan Yang, Furu Wei, Tao Yu, Amanpreet Singh, and Douwe Kiela. 2024. Generative representational instruction tuning. *arXiv preprint arXiv:2402.09906*.

Arvind Neelakantan, Tao Xu, Raul Puri, Alec Radford, Jesse Michael Han, Jerry Tworek, Qiming Yuan, Nikolas Tezak, Jong Wook Kim, Chris Hallacy, and 1 others. 2022. Text and code embeddings by contrastive pre-training. *arXiv preprint arXiv:2201.10005*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *EMNLP-IJCNLP 2019*, pages 3982–3992.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning to retrieve prompts for in-context learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2655–2671.

Oscar Sainz, Oier Lopez de Lacalle, Gorka Labaka, Ander Barrena, and Eneko Agirre. 2021. Label verbalization and entailment for effective zero and few-shot relation extraction. In *EMNLP 2021*, pages 1199–1212.

Oscar Sainz, Iker García-Ferrero, Rodrigo Agerri, Oier Lopez de Lacalle, German Rigau, and Eneko Agirre. 2024. Gollie: Annotation guidelines improve zero-shot information-extraction. In *ICLR 2024*.

Jacob Mitchell Springer, Suhas Kotha, Daniel Fried, Graham Neubig, and Aditi Raghunathan. 2024. Repetition improves language model embeddings. *arXiv preprint arXiv:2402.15449*.

George Stoica, Emmanouil Antonios Platanios, and Barnabás Póczos. 2021. Re-tacred: Addressing shortcomings of the tacred dataset. In *Proceedings of the AAAI conference on artificial intelligence*, pages 13843–13850.

Qi Sun, Kun Huang, Xiaocui Yang, Rong Tong, Kun Zhang, and Soujanya Poria. 2024. Consistency guided knowledge retrieval and denoising in llms for zero-shot document-level relation triplet extraction. In *Proceedings of the ACM on Web Conference 2024*, pages 4407–4416.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv: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 *EMNLP 2023*, pages 3534–3547.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2023. Improving text embeddings with large language models. *arXiv preprint arXiv:2401.00368*.

Shanchan Wu and Yifan He. 2019. Enriching pre-trained language model with entity information for relation classification. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 2361–2364.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighof. 2023. C-pack: Packaged resources to advance general chinese embedding. *arXiv preprint arXiv:2309.07597*.

Kaijia Yang, Nantao Zheng, Xinyu Dai, Liang He, Shujian Huang, and Jiajun Chen. 2020. Enhance prototypical network with text descriptions for few-shot relation classification. In *Proceedings of the 29th ACM*

International Conference on Information & Knowledge Management, pages 2273–2276.

Jinsung Yoon, Sercan O Arik, Yanfei Chen, and Tomas Pfister. 2023. Search-adaptor: Text embedding customization for information retrieval. *arXiv preprint arXiv:2310.08750*.

Kai Zhang, Bernal Jiménez Gutiérrez, 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*, pages 794–812.

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*, pages 35–45.

Jun Zhao, Wenyu Zhan, Wayne Xin Zhao, Qi Zhang, Tao Gui, Zhongyu Wei, Junzhe Wang, Minlong Peng, and Mingming Sun. 2023a. Re-matching: A fine-grained semantic matching method for zero-shot relation extraction. In *ACL 2023*, pages 6680–6691.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, and 1 others. 2023b. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. 2024. Large language models are not robust multiple choice selectors. In *ICLR 2024*.

Hengyi Zheng, Rui Wen, Xi Chen, Yifan Yang, Yunyan Zhang, Ziheng Zhang, Ningyu Zhang, Bin Qin, Xu Ming, and Yefeng Zheng. 2021. Prgc: Potential relation and global correspondence based joint relational triple extraction. In *ACL-IJCNLP 2021*, pages 6225–6235.

A Details of Generation Augmentation

We provide several examples of running *Sentence Distillation* and *Relation Enrichment* on WikiZSL in Tables 4 and 5, respectively. These examples all run on *GPT-3.5-turbo-0125*. It is noteworthy that for all LLMs, the JSON parsing rate can reach over 99.2%. For responses that could not be parsed, we directly use the sentences and entities from the raw data as the "*relation_sentence*" and "*entity_type*."

B Details of VanillaPrompt and QA4RE

To investigate the performance of VanillaPrompt and QA4RE on Wiki-ZSL and FewRel, we follow the approach of Zhang et al. (2023), manually crafting templates for all relations. Examples of VanillaPrompt and QA4RE running on *GPT-3.5-turbo-0125* are provided in Table 6. It is worth noting that the quality of manually constructed templates may affect performance, thus the results we report may contain slight errors.

C Error Analysis

In Figure 8, we report the accuracy of our framework across 80 categories of relations in FewRel. We observe that, after plugging the retrieval adjuster, the retrieval performance for most relation categories improved (indicated in green). For instance, similar relations such as "*P131: located in the administrative territorial entity*," "*P276: location*," "*P740: location of formation*," and "*P937: work location*" can often be confused during retrieval. The retrieval adjuster can learn the relation patterns from seen data and effectively distinguish between similar relations. However, the retrieval adjuster might decrease the performance for a few relations (indicated in red). For example, for the relation "*P400: platform*," some instances might be predicted as "*P710: participant*" after adjustment. Optimizing the retrieval adjuster to address these challenges will be part of our future work.

Additionally, we observe that our framework performs poorly on the relation "*P750: distributor*" because these instances are often misclassified as other relations such as "*P123: publisher*" or "*P57: director*." We suspect this happens because they are frequently mentioned in similar contexts. (e.g., in discussions about the commercialization of films or TV shows, these relation patterns might coexist). Therefore, due to this co-occurrence pattern, LLMs may easily focus on incorrect information during sentence distillation or semantic representation, leading to failures. Therefore, we believe that further enriching the semantics of relation labels and improving their distinctiveness will be the direction for future optimization.

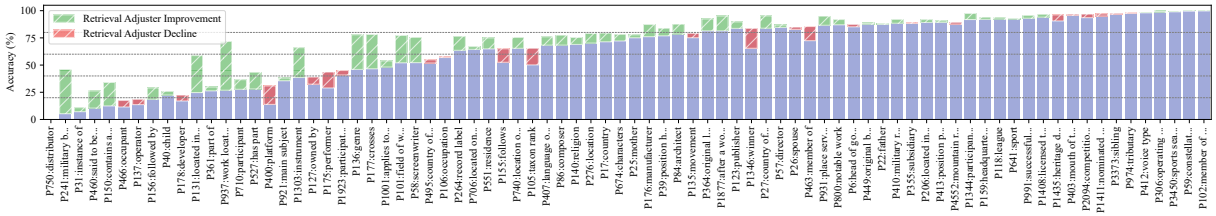

Figure 8: We conduct 10 runs on the FewRel dataset with the setting of $m = 15$. Each relation is covered at least once in the unseen relations set R_u. We obtain the average accuracy across all 80 relations. Green and red represent the performance increase or decrease on each relation after adding the retrieval adjuster, respectively.

Examples of Sentence Distillation
Prompt: Provide a summary of the relation between head entity "**{Head Entity}**" and tail entity "**{Tail Entity}**" in one sentence. Make sure that these are the only two entities in this sentence, and the head entity is in front of the tail entity. Give the entity types of the head entity and tail entity, and finally present them in JSON format. Desired format:{"relation_sentence":"","head_type":"","tail_type":""} Text:"""**{Sentence}**"""
Input: **{Sentence}:** Robin Rand (born c. 1956) is a United States Air Force general and commander of Air Force Global Strike Command. **{Head Entity}:** Air Force Global Strike Command. **{Tail Entity}:** United States Air Force. ***Output:*** {"relation_sentence":"Air Force Global Strike Command is commanded by a general from the United States Air Force.", "head_type":"Military Unit", "tail_type":"Armed Forces"}.
Input: **{Sentence}:** Eye to Ear II is a studio album by English guitarist , composer and improvisor Fred Frith. **{Head Entity}:** Fred Frith. **{Tail Entity}:** composer. ***Output:*** {"relation_sentence":"Fred Frith is a composer.", "head_type":"Person", "tail_type":"Occupation"}.
Input: **{Sentence}:** Guy Rodgers from Temple University was selected before the draft as Philadelphia Warriors territorial pick. **{Head Entity}:** Guy Rodgers. **{Tail Entity}:** Philadelphia Warriors. ***Output:*** {"relation_sentence":"Guy Rodgers was selected as a territorial pick by the Philadelphia Warriors.", "head_type":"Person", "tail_type":"Sports Team"}.
Input: **{Sentence}:** His grandfather Uma Shankar Dikshit was a part of the freedom struggle for thirty years who was closely associated with Pandit Jawahar Lal Nehru , Indira Gandhi and Rajiv Gandhi. **{Head Entity}:** Rajiv Gandhi. **{Tail Entity}:** Indira Gandhi. ***Output:*** {"relation_sentence":"Rajiv Gandhi is the son of Indira Gandhi.", "head_type":"Person", "tail_type":"Person"}.
Input: **{Sentence}:** Developed by SanDisk, the Cruzer Enterprise is an encrypted USB flash drive. **{Head Entity}:** SanDisk. **{Tail Entity}:** USB flash drive. ***Output:*** {"relation_sentence":"SanDisk developed the Cruzer Enterprise.", "head_type":"Company", "tail_type":"Product"}.

Table 4: Examples of Sentence Distillation on WikiZSL.

Examples of Relation Enrichment
Prompt: Please provide all possible types of head entity and tail entity for the given relation, and finally present them in JSON format. Desired format: {"relation":"", "head_types":[], "tail_types":[]} Relation:"""**{Label}**: **{Description}**"""
Input: **{Label}:** country of citizenship. **{Description}:** the object is a country that recognizes the subject as its citizen ***Output:*** {"relation": "country of citizenship", "head_types":["Person"], "tail_types":["Country"]}
Input: **{Label}:** director. **{Description}:** director(s) of film, TV-series, stageplay, video game or similar ***Output:*** {"relation": "director", "head_types":["Film", "TV-series", "Stageplay", "Video game"], "tail_types":["Person"]}
Input: **{Label}:** voice type. **{Description}:** person's voice type. expected values: soprano, mezzo-soprano, contralto, countertenor, tenor, baritone, bass (and derivatives) ***Output:*** {"relation": "voice type", "head_types":["person"], "tail_types":["soprano", "mezzo-soprano", "contralto", "countertenor", "tenor", "baritone", "bass"]}
Input: **{Label}:** conflict. **{Description}:** battles, wars or other military engagements in which the person or item participated ***Output:*** {"relation": "conflict", "head_types":["person", "organization"], "tail_types":["battle", "war", "military engagement"]}
Input: **{Label}:** said to be the same as. **{Description}:** this item is said to be the same as that item, but the statement is disputed ***Output:*** {"relation": "said to be the same as", "head_types":["Concept", "Object"], "tail_types":["Concept", "Object"]}

Table 5: Examples of Relation Enrichment on WikiZSL.

Example of VanillaPrompt

Input:

Given a sentence and two entities within the sentence, classify the relationship between the two entities based on the provided sentence. Please give the relationship only. All possible relationships are listed below:
- architect
- main subject
- winner
- said to be the same as
- participant
- nominated for
- field of work
- place served by transport hub
- tributary
- military rank
- league
- heritage designation
- member of political party
- location of formation
- follows

Sentence: California longhorn cattle were given as a gift to Hawaiian King Kamehameha I by British Captain George Vancouver in 1793.
Entity 1: george vancouver
Entity 2: captain
Relationship:

Output:

Entity 1: George Vancouver
Entity 2: Captain
Relationship: Military rank

Example of QA4RE

Input:

Determine which option can be inferred from the given Sentence. You can only choose one of the following options.
Sentence: California longhorn cattle were given as a gift to Hawaiian King Kamehameha I by British Captain George Vancouver in 1793 .
Options:
A. [george vancouver] is the architect of [captain]
B. [george vancouver] is the main subject of [captain]
C. [george vancouver] is the winner of [captain]
D. [george vancouver] is said to be the same as [captain]
E. [george vancouver] is a participant in [captain]
F. [george vancouver] was nominated for [captain]
G. [george vancouver] works in the field of [captain]
H. [george vancouver] serves as the place served by [captain] transport hub
I. [george vancouver] is a tributary of [captain]
J. [george vancouver] holds the military rank of [captain]
K. [george vancouver] is a member of the league [captain]
L. [george vancouver] has the heritage designation [captain]
M. [george vancouver] is a member of the political party [captain]
N. [george vancouver] is the location where [captain] was formed
O. [george vancouver] follows [captain]
Which option can be inferred from the given sentence?
Option:

Output:

J

Table 6: Several running examples of VanillaPrompt and QA4RE based on the descriptions in the original paper (Zhang et al., 2023).