LLM-OREF: An Open Relation Extraction Framework Based on Large Language Models

Hongyao Tu[1,3*†] **Liang Zhang**[1*] **Yujie Lin**[1] **Xin Lin**[3]
Haibo Zhang[2] **Long Zhang**[2] **Jinsong Su**[1,3‡]
[1]School of Informatics, Xiamen University, China,
[2]LLM Team, Shopee Pte. Ltd.
[3]National Institute for Data Science in Health and Medicine, Xiamen University
{tuhongyao,lzhang}@stu.xmu.edu.cn, jssu@xmu.edu.cn

Abstract

The goal of open relation extraction (OpenRE) is to develop an RE model that can generalize to new relations not encountered during training. Existing studies primarily formulate OpenRE as a clustering task. They first cluster all test instances based on the similarity between the instances, and then manually assign a new relation to each cluster. However, their reliance on human annotation limits their practicality. In this paper, we propose an OpenRE framework based on large language models (LLMs), which directly predicts new relations for test instances by leveraging their strong language understanding and generation abilities, without human intervention. Specifically, our framework consists of two core components: (1) a relation discoverer (RD), designed to predict new relations for test instances based on *demonstrations* formed by training instances with known relations; and (2) a relation predictor (RP), used to select the most likely relation for a test instance from n candidate relations, guided by *demonstrations* composed of their instances. To enhance the ability of our framework to predict new relations, we design a self-correcting inference strategy composed of three stages: relation discovery, relation denoising, and relation prediction. In the first stage, we use RD to preliminarily predict new relations for all test instances. Next, we apply RP to select some high-reliability test instances for each new relation from the prediction results of RD through a cross-validation method. During the third stage, we employ RP to re-predict the relations of all test instances based on the demonstrations constructed from these reliable test instances. Extensive experiments on three OpenRE datasets demonstrate the effectiveness of our framework. We release our code at https://github.com/XMUDeepLIT/LLM-OREF.git.

1 Introduction

Relation Extraction (RE), as a crucial task in information extraction, aims to extract relations between entity pairs from unstructured text. The extracted relations play a vital role in many downstream applications, such as search engine (Li et al., 2006), question answering (Yu et al., 2017), and knowledge base population (Ji and Grishman, 2011). Conventional RE studies mainly focus on building models that can only handle predefined relations, inherently limiting their utility in real-world scenarios where new relations continually emerge. To address this limitation, researchers have turned to Open Relation Extraction (OpenRE), which is not confined to a predefined set of relations and can dynamically discover new ones, making it more practical for real-world applications.

In this regard, dominant methods formulate OpenRE as a clustering task (Yao et al., 2011; Marcheggiani and Titov, 2016; Elsahar et al., 2017), which aggregates semantically related relation instances into the same cluster, with each cluster representing a potential new relation. Along this line, subsequent studies directly utilize pretrained language models (e.g., BERT (Devlin et al., 2019)) to encode an instance for obtaining its relational representation and then perform clustering on these representations (Hu et al., 2020). Since pretrained language models have not been fine-tuned on RE datasets, the performance of such methods remains suboptimal. To deal with this issue, several methods leverage the available labeled datasets for RE (which only contain known relations) to fine-tune pretrained language models (Zhao et al., 2021; Wang et al., 2022). However, the above methods cannot align clusters with specific relation types, restricting their applicability in downstream tasks. While Zhao et al. (2023) mitigates this issue by actively selecting representative instances for human annotation during clustering, their approach

[*] Equal contribution.

[†] This work was partially done while Hongyao Tu was interning at the Shopee LLM Team.

[‡] Corresponding author.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 9040–9052
November 4-9, 2025 ©2025 Association for Computational Linguistics

remains impractical for real-world deployment due to its reliance on human intervention.

Recently, Large Language Models (LLMs) have demonstrated strong text understanding capabilities across various downstream tasks and can effectively capture complex relation patterns in text sequences (Wan et al., 2023; Wadhwa et al., 2023). More importantly, unlike traditional classification and clustering models, the generative nature of LLMs allows them to predict new relations in the form of natural language directly. Therefore, we believe that exploring the potential of LLMs in OpenRE is a promising research direction.

In this work, we conduct a preliminary study to investigate the capabilities of LLMs in OpenRE. We first observe that LLMs perform poorly at predicting new relations in a zero-shot manner. In addition, providing LLMs with a few-shot demonstration that includes instances that do not belong to new relations can improve their performance, but the improvement is limited. Interestingly, we find that their performance significantly improves when instances of the new relations are provided in few-shot demonstrations. These findings suggest that while LLMs excel in comprehending relations through demonstrations, they still struggle to discover new relations.

Based on these insights, we propose an LLM-based Open Relation Extraction Framework (LLM-OREF), which consists of two key components: the Relation Discoverer (RD) and the Relation Predictor (RP). The former aims to preliminarily predict new relations for test instances by capturing relation patterns from demonstrations composed of instances with known relations. The latter is designed to deeply understand the relations of instances in demonstration, and then accurately determine the most probable relation for the test instance from these relations. The primary distinction between RD and RP lies in their input: whether the demonstration contains the relation of the test instance. To reduce storage and training costs, both RD and RP are built on the same LLM using the LoRA (Hu et al., 2022) fine-tuning strategy. During training, since we can only access instances of known relations, we construct the inputs of RD and RP using these instances for corresponding training. As RP's demonstration includes the target relation of the test instance, which greatly reduces the difficulty of relation prediction, its performance is significantly better than that of RD. Therefore, to enhance the training of RD, we introduce an extra distillation loss ($\mathcal{L}_{\text{KD}}$) in its training objective, designed to leverage the output distribution of RP to guide RD's training.

To more effectively coordinate RD and RP to discover new relations in real-world scenarios and accurately predict relations of test instances, we propose a self-correcting inference strategy. In particular, the strategy involves three stages: relation discovery, relation denoising, and relation prediction. In the first stage, we employ RD to initially predict new relations for each instance in the test set, based on demonstrations consisting of training instances with known relations. In the second stage, considering that RD's predictions may contain noise, we use RP to cross-validate the accuracy of the predicted relation for each test instance, yielding a set of reliable instances for each new relation. In the third stage, we apply RP to more accurately predict new relations for each test instance using demonstrations constructed from these reliable instances.

To summarize, the main contributions of this work are as follows: (1) We propose LLM-OREF, a novel OpenRE framework based on LLMs that includes two key components, the RD and RP, to enable the discovery of new relations and their accurate prediction. (2) We propose a self-correcting inference strategy that progressively refines new relation prediction through a three-stage pipeline of relation discovery, relation denoising, and relation prediction. (3) Extensive experiments conducted on three OpenRE datasets demonstrate the effectiveness of our framework.

2 Preliminary Study

In this section, we conduct a preliminary study to explore the ability of LLMs in discovering new relations. To this end, we evaluate the performance of an open-source LLM, LLaMA-2-7B (Touvron et al., 2023), on a commonly used OpenRE dataset FewRel (Han et al., 2018).

Specifically, we first simply evaluate the accuracy of the LLM on the test set of FewRel under a zero-shot setting. The red dashed line in Figure 1 indicates that the LLM exhibits notably poor performance. This is mainly attributed to the fact that the model lacks task-specific guidance, having not been exposed to any examples that illustrate the structure and requirements of RE. To enhance the LLM's understanding of the RE task, we provide it with demonstrations, including instances of known

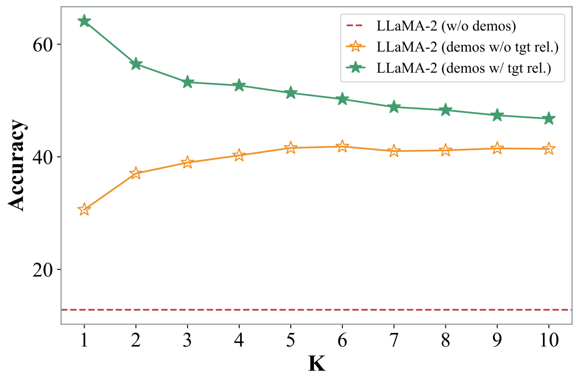

Figure 1: Accuracy of **LLaMA-2-7B** on FewRel. "***w/o demons***" means no demonstrations are given; "***demos w/o tgt rel.***" and "***demos w/ tgt rel.***" refer to demonstrations that do not contain and that contain instances sharing the target relation of the test instance, respectively. On the x-axis, "***K***" denotes the number of instances in demonstrations. In the "demos w/ tgt rel." setting, as K increases, it becomes harder for the model to identify the target relation, leading to a gradual drop in accuracy.

relations in the training set of FewRel, to predict new relations of test instances under a few-shot setting. While this setting improves the LLM's ability to identify new relations, its performance is still inadequate for real-world applications (see yellow-☆ line in Figure 1). These results intuitively reveal that the ability of LLMs to discover new relations is still limited. This motivates us to further explore more effective methods and strategies to enhance the performance of LLMs on OpenRE.

Furthermore, we explore the performance of LLMs when demonstrations include the target (new) relations of test instances, as done in standard in-context learning (ICL) (Brown et al., 2020). From the green-★ line in Figure 1, we observe a substantial improvement in the LLM's performance. These findings indicate that LLMs can effectively grasp the semantics of a relation through its instances in the demonstration, enabling them to accurately identify the most likely relation for a test instance from those presented in the demonstration. This has been noted in previous ICL-based RE studies (Wan et al., 2023; Rajpoot and Parikh, 2023) and inspired the design of our framework.

3 Problem Definition

OpenRE endeavors to accurately predict target (new) relations for test instances in real-world scenarios. Hence, given an unlabeled dataset $D{=}\{x_i\}_{i=1}^{N}$ (i.e., test set) with N test instances, the OpenRE model is required to predict a new relation y_i for each test instance x_i. Meanwhile, each instance $x_i{=}{<}s_i, h_i, t_i{>}$ consists of a sentence s_i, a head entity h_i, and a tail entity t_i. Following recent works (Zhao et al., 2021; Wang et al., 2022), we use a training set $D'{=}\{x'_j\}_{j=1}^{M}$ containing M instances of known relations to adapt LLMs to the RE task. Here, each training instance $x'_j{=}{<}s'_j, h'_j, t'_j{>}$ is annotated with its associated relation label y'_j. Notably, in OpenRE, the relations in the test set do not overlap with those in the training set.

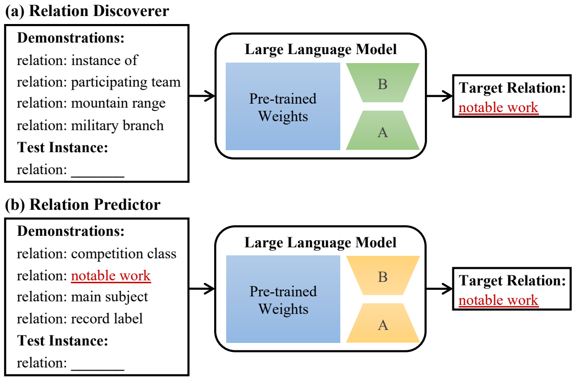

Figure 2: Illustration of two key components in our framework. The demonstrations in RD consist of known relation instances, while the demonstrations in RP are composed of new relation instances. Additionally, the demonstrations in RD do not include the target relation of the test instance, whereas RP does. Both RD and RP are built on the same LLM using the LoRA fine-tuning strategy.

4 Our Framework

In this section, we provide a detailed description of our LLM-based OpenRE framework, LLM-OREF. In the following, we first elaborate on two key components of LLM-OREF in §4.1, and then detail the training and inference strategies of our framework in §4.2 and §4.3, respectively.

4.1 Overall framework

As illustrated in Figure 2, LLM-OREF consists of two key components: the Relation Discoverer and the Relation Predictor.

Relation Discoverer (RD). The RD endeavors to predict new relations for test instances based on demonstrations consisting of instances with known relations.

As illustrated in Figure 2 (a), for each test instance x_i, we first randomly sample n known relations from the training set, each associated with a training instance, and concatenate them to form demonstrations: $D_{\text{RD}}{=}[x'_1, y'_1, ..., x'_{\text{n}}, y'_{\text{n}}]$. Subsequently, we construct the input I_{RD} for RD by con-

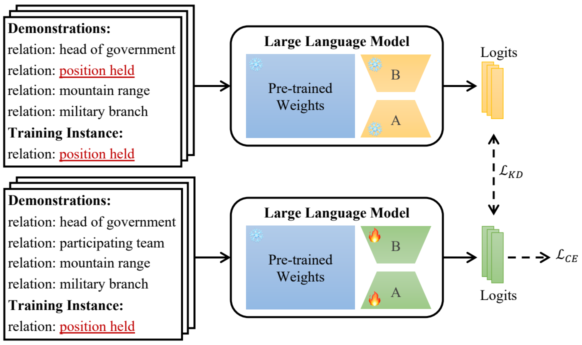

Figure 3: Illustration of the training strategy for RD. Since RP's demonstration includes the target relation, relation prediction becomes easier, resulting in better performance than RD. To improve RD, we introduce a distillation loss ($\mathcal{L}_{\text{KD}}$) that leverages RP's output distribution to guide RD's training.

catenating the instruction prompt P_{RD}, the demonstrations D_{RD}, and the test instance x_i, forming I_{RD}=$[P_{\text{RD}}; D_{\text{RD}}; x_i]$. Here, P_{RD} serves to guide the RD in comprehensively understanding the RE task through the demonstrations D_{RD}, while also instructing it to predict a new relation (not contained in D_{RD}) for the test instance x_i. Finally, the RD takes I_{RD} as input to autoregressively generate a new relation $\hat{y}_i$ for x_i.

Relation Predictor (RP). As the discovery of new relations is inherently challenging, the RD may produce noisy predictions. To mitigate this, the RP is employed to denoise and refine these predictions for the test instances.

Specifically, as shown in Figure 2 (b), for each x_i with its predicted relation $\hat{y}_i$, we first randomly select $n{-}1$ new relations from the prediction results of RD on the test set, each accompanied by a test instance, and additionally sample a test instance belonging to $\hat{y}_i$. Next, these instances are concatenated to form demonstrations D_{RP}=$[x_1, \hat{y}_1, ..., x_n, \hat{y}_n]$. Then, we create an input I_{RP}=$[P_{\text{RP}}; D_{\text{RP}}; x_i]$ for RP by concatenating a specific instruction prompt P_{RP}, the demonstration D_{RP}, and the test instance x_i, where P_{RP} instructs RP to identify the most likely relation for x_i from those contained in D_{RP}. Finally, we input I_{RP} into RP to obtain a new predicted relation $\tilde{y}_i$ for x_i, which can be used to verify or refine the initial prediction $\hat{y}_i$ provided by RD.

4.2 Model Training

To effectively train both RD and RP in our framework, we adopt a two-stage training strategy. For storage efficiency, we adopt LoRA to fine-tune a shared LLM for both RP and RD. Notably, during OpenRE training, the model is trained on the training set D', which only contains instances of known relations. Thus, the inputs for training RP and RD are solely constructed from relations and instances sampled within D'.

The first stage. Here, we focus on effectively training RP using the training set D'. Specifically, for each training instance x'_j with its corresponding relation y'_j, we first randomly sample corresponding demonstrations D_{RP} from D'. According to the objective of RP, the demonstrations D_{RP} are required to include instances belonging to the relation y'_j. Then, we construct the input I_{RP}=$[P_{\text{RP}}; D_{\text{RP}}; x'_j]$ and feed it into RP. Finally, we train RP to autoregressively generate the relation y'_j for the training instance x'_j using the cross-entropy loss $\mathcal{L}_{\text{CE}}$:

$$\mathcal{L}_{\text{CE}} = -\frac{1}{|y'_j|}\sum_{t=1}^{|y'_j|} \log P_{\boldsymbol{\theta}}(y'^{(t)}_j \mid y'^{(<t)}_j, \mathbf{I_{RP}}), \quad (1)$$

where $\boldsymbol{\theta}$ denotes the learnable LoRA parameters for RP, and t is the index of a token in y'_j.

The second stage. At this stage, we aim to enhance RD's ability to discover new relations. For each training instance x'_j labeled with relation y'_j, we construct an RD's input $I_{\text{RD}} = [P_{\text{RD}}; D_{\text{RD}}; x'_j]$, ensuring that the demonstration D_{RD}, sampled from D', does not include any instances of y'_j. Next, we input I_{RD} into RD and compute the autoregressive loss $\mathcal{L}'_{\text{CE}}$ with respect to the target relation y'_j:

$$\mathcal{L'}_{\text{CE}} = -\frac{1}{|y'_j|}\sum_{t=1}^{|y'_j|} \log P_{\boldsymbol{\phi}}(y'^{(t)}_j \mid y'^{(<t)}_j, \mathbf{I_{RD}}), \quad (2)$$

where $\boldsymbol{\phi}$ is the learnable LoRA parameters of RD.

Since D_{RP} contains instances of the target relation y'_j, whereas D_{RD} does not, RP can more easily predict the relation x'_j than RD. Therefore, we also employ the RP trained in the first stage as a teacher to guide the training of RD. As depicted in Figure 3, we compute the KL divergence $\mathcal{L}_{\text{KD}}$ between the predictive distributions of RP and RD for the same training instance x'_j:

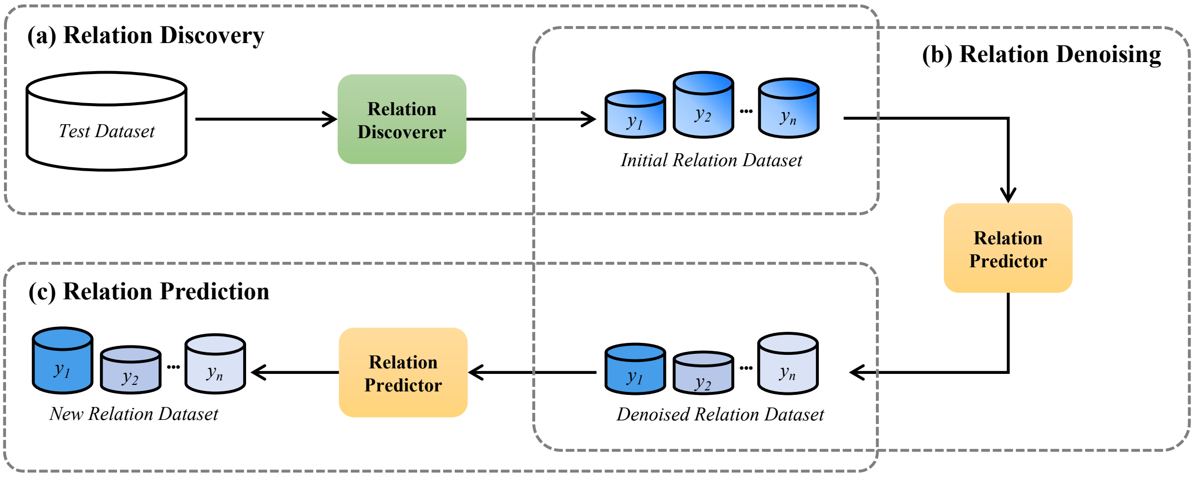

Figure 4: The illustration of our self-correcting inference strategy. It consists of three stages: (a) relation discovery, where the RD discovers potential new relations of test instances; (b) relation denoising, where the RP identifies high-reliability instances for each new relation from the prediction results of RD; and (c) relation prediction, using high-reliability instances of new relations to construct demonstrations, so that the RP can better predict relations of test instances.

$$\begin{aligned}\mathcal{L}_{\text{KD}} = \frac{1}{|y'_j|}\sum_{t=1}^{|y'_j|}\mathbf{KL}&\Big(P_{\boldsymbol{\theta}}(y'^{(t)}_j \mid y'^{(<t)}_j, \mathbf{I_{RP}}) \\ &||P_{\phi}(y'^{(t)}_j \mid y'^{(<t)}_j, \mathbf{I_{RD}})\Big)\,.\end{aligned} \tag{3}$$

Finally, the training objective for RD is given by $\mathcal{L}_{\text{RD}} = \mathcal{L}'_{\text{CE}} + \alpha\mathcal{L}_{\text{KD}}$, where α is a hyperparameter used to balance the impact of $\mathcal{L}_{\text{KD}}$ on RD training.

4.3 Self-Correcting Inference Strategy

To coordinate RD and RP to effectively discover new relations and accurately predict the relation for each test instance, we propose a self-correcting inference strategy. As depicted in Figure 4, this strategy consists of three stages: relation discovery, relation denoising, and relation prediction.

Relation Discovery. During this phase, RD is used to perform initial predictions of new relations for all instances in the test set D.

Specifically, for each test instance x_i, we first construct RD's input I_{RD} by randomly sampling instances with known relations from the training set D' to form the corresponding demonstration D_{RD}. Then, we feed the input I_{RD} into RD to obtain the predicted relation $\hat{y}_i$ for x_i. Furthermore, to improve the recall of RD in discovering new relations, we make multiple predictions for each test instance x_i using different demonstrations to obtain multiple prediction relations $[\hat{y}_i^1, ..., \hat{y}_i^K]$ for each test instance x_i, where K is the number of predictions.

Relation Denoising. Here, we focus on using RP to pick out some high-reliability samples for each new relation from the prediction results of RD.

Given each test instance x_i and its predicted relation $\hat{y}_i^k$, we first sample multiple demonstrations $[D_{\text{RP}}^1, ..., D_{\text{RP}}^d]$, each comprising $\hat{y}_i^k$ and other new relations. Notably, we ensure that these demonstrations cover all new relations discovered in the previous stage. Subsequently, we utilize these sampled demonstrations to build the corresponding input $[I_{\text{RP}}^1, ..., I_{\text{RP}}^d]$ for the test instance x_i. Next, we feed these inputs into RP to generate new predictions for x_i. This allows us to assess the reliability of $\hat{y}_i^k$ by comparing it to other new relations. If RP consistently outputs $\hat{y}_i^k$ across multiple predictions, we consider it a reliable relation for the test instance x_i.

After denoising all test instances, the remaining ones undergo further rounds of denoising. In total, we perform T rounds to obtain high-reliability samples.

Relation Prediction. Following the prior phase, we gathered the reliable test instances for each new relation. In this stage, we utilize these reliable test instances to construct demonstrations D_{RP} that enable RP to precisely predict new relations of other test instances.

Specifically, for a test instance x_i, we first sample n new relations along with their reliable test instances to construct a demonstration D_{RP}. We then apply RP to select the most probable relation $\hat{y}_i$ for x_i from these n candidates. After obtaining

$\hat{y}_i$, we sample other $n-1$ new relations and include $\hat{y}_i$ to form the candidate set for the next prediction. We repeat this process until all new relations have been traversed, ultimately obtaining the most reliable relation for x_i.

5 Experiments

5.1 Datasets & Evaluation Metrics

Following Zhao et al. (2023), we conduct experiments on two widely used RE datasets: FewRel (Han et al., 2018) and TACRED (Zhang et al., 2017), as well as a constructed RE dataset, FewRel-LT (Zhao et al., 2023). For each dataset, we split the relation types into disjoint sets of known and new relations. The details of datasets are in **Appendix** A.1.

Following Zhao et al. (2023), we use B^3 (Bagga and Baldwin, 1998), V-measure (Rosenberg and Hirschberg, 2007), Adjusted Rand Index (ARI) (Hubert and Arabie, 1985), and Macro-F_1 (Opitz and Burst, 2019) to measure the precision and recall of results, homogeneity and completeness of results, the agreement between results and true distributions, and the classification performance.

5.2 Baselines

We compare our LLM-OREF (based on **LLaMA-2-7B** (Touvron et al., 2023) and **Qwen2.5-14B** (Yang et al., 2025)) with two vanilla models and five representative OpenRE baselines, including: 1) **RW-HAC** (Elsahar et al., 2017), 2) **SelfORE** (Hu et al., 2020), 3) **RSN** (Wu et al., 2019), 4) **RoCORE** (Zhao et al., 2021), 5) **ASCORE** (Zhao et al., 2023). The details of these baselines are in **Appendix** A.2.

5.3 Implementation Details

For all experiments, the number of demonstrations n is set to 4, and the number of relation discovery predictions K for each test instance is set to 3. The number of relation denoising iterations is set to T=3. The prompt templates for RD and RP are in **Appendix** A.3. When training, we adopt AdamW (Loshchilov and Hutter) as the optimizer, along with a linear learning rate schedule. All models are trained for one epoch using LoRA with r=64 and α=64. The hyper-parameters, including the learning rates for RD and RP, the distillation temperature, and the corresponding loss weight α, are listed in Table 2. During inference, we utilize vLLM (Kwon et al., 2023) for efficient inference acceleration. All experiments are conducted on two NVIDIA H100 (80GB).

5.4 Main Results

Table 1 presents the main experimental results comparing our framework with a range of baselines across three datasets. Next, we provide a detailed analysis of the results:

The results in Table 1 show that our framework achieves superior performance on both the class-balanced dataset FewRel and the class-imbalanced datasets TACRED and FewRel-LT. It consistently outperforms all baselines, including the state-of-the-art ASCORE, while eliminating the need for human intervention. These results not only demonstrate the effectiveness of our framework but also provide valuable insights for future research in OpenRE.

Compared with traditional clustering methods such as RW-HAC, SelfORE, RSN, and RoCORE, which cannot automatically align clusters with specific relations, our framework demonstrates markedly greater practicality. It consistently outperforms these baselines across all datasets and achieves particularly large performance gains on the challenging TACRED and FewRel-LT datasets, both of which suffer from severe class imbalance. This superior performance under imbalanced conditions further validates the real-world applicability of our framework.

From Table 1, we observe that the vanilla LLaMA-2-7B and Qwen2.5-14B exhibit consistently poor performance across all datasets, underscoring the limitations of directly applying LLMs for OpenRE. In contrast, our framework, built upon these models, achieves significant and consistent improvements, highlighting the necessity of a tailored framework for adapting LLMs to OpenRE. Moreover, these results demonstrate that our framework can be effectively applied to different LLMs, further confirming its generality.

5.5 Ablation Study

We further conduct ablation studies by removing various components of our framework to assess their individual contributions. Specifically, we compare our framework with the following variants in Table 3.

(1) *w/o. self-correcting inference strategy.* In this variant, we remove the self-correcting inference strategy from LLM-OREF, using RD for new relation discovery, which is then taken as the final

Dataset	Method	B^3			V-measure			ARI	Classification		
		Prec.	Rec.	F_1	Hom.	Comp.	F_1		Prec.	Rec.	F_1
FewRel	RW-HAC (Elsahar et al., 2017)	0.175	0.367	0.237	0.357	0.463	0.403	0.108	0.251	0.264	0.216
	SelfORE (Hu et al., 2020)	0.527	0.552	0.539	0.728	0.736	0.732	0.517	0.604	0.632	0.600
	RSN (Wu et al., 2019)	0.174	0.640	0.274	0.389	0.659	0.489	0.173	0.112	0.239	0.134
	RoCORE (Zhao et al., 2021)	0.806	0.843	0.824	0.883	0.896	0.889	0.807	0.827	0.868	0.837
	ASCORE (Zhao et al., 2023)	0.799	0.841	0.820	0.888	0.901	0.894	0.801	0.832	0.862	0.838
	LLaMA-2-7B	0.528	0.327	0.404	0.694	0.527	0.599	0.373	0.608	0.402	0.430
	Qwen2.5-14B	0.546	0.555	0.550	0.725	0.712	0.719	0.485	0.710	0.596	0.586
	Ours (LLaMA-2-7B)	0.647	0.700	0.672	0.790	0.809	0.800	0.637	0.750	0.737	0.718
	Ours (Qwen2.5-14B)	0.817	0.850	**0.833**	0.893	0.905	**0.899**	**0.810**	0.887	0.883	**0.879**
TACRED	RW-HAC (Elsahar et al., 2017)	0.317	0.668	0.430	0.443	0.668	0.532	0.291	0.244	0.246	0.171
	SelfORE (Hu et al., 2020)	0.517	0.441	0.476	0.631	0.600	0.615	0.434	0.343	0.396	0.360
	RSN (Wu et al., 2019)	0.312	0.807	0.451	0.445	0.768	0.563	0.354	0.149	0.118	0.225
	RoCORE (Zhao et al., 2021)	0.696	0.685	0.690	0.786	0.786	0.787	0.640	0.547	0.594	0.563
	ASCORE (Zhao et al., 2023)	0.742	0.821	0.780	0.807	0.856	0.831	0.781	0.698	0.715	0.699
	LLaMA-2-7B	0.441	0.305	0.361	0.474	0.346	0.400	0.159	0.377	0.450	0.325
	Qwen2.5-14B	0.683	0.610	0.644	0.713	0.656	0.684	0.619	0.709	0.648	0.592
	Ours (LLaMA-2-7B)	0.739	0.700	0.719	0.769	0.731	0.749	0.798	0.665	0.742	0.633
	Ours (Qwen2.5-14B)	0.803	0.775	**0.789**	0.817	0.858	**0.837**	**0.893**	0.713	0.784	**0.704**
FewRel-LT	RW-HAC (Elsahar et al., 2017)	0.255	0.322	0.285	0.379	0.421	0.399	0.145	0.190	0.176	0.160
	SelfORE (Hu et al., 2020)	0.563	0.456	0.504	0.717	0.661	0.687	0.377	0.439	0.526	0.462
	RSN (Wu et al., 2019)	0.211	0.500	0.297	0.350	0.510	0.415	0.193	0.098	0.173	0.117
	RoCORE (Zhao et al., 2021)	0.662	0.717	0.689	0.800	0.801	0.800	0.581	0.507	0.538	0.517
	ASCORE (Zhao et al., 2023)	0.650	0.845	0.735	0.790	0.885	0.835	0.676	0.530	0.609	0.550
	LLaMA-2-7B	0.588	0.291	0.389	0.725	0.511	0.600	0.269	0.549	0.404	0.412
	Qwen2.5-14B	0.601	0.516	0.555	0.743	0.679	0.710	0.478	0.665	0.595	0.547
	Ours (LLaMA-2-7B)	0.651	0.655	0.653	0.778	0.767	0.773	0.594	0.713	0.736	0.698
	Ours (Qwen2.5-14B)	0.777	0.775	**0.776**	0.862	0.858	**0.860**	**0.738**	0.856	0.876	**0.855**

Table 1: Main results on three OpenRE datasets. The experimental results demonstrate the effectiveness of our framework under both class-balanced and class-imbalanced settings.

Dataset	Model	RP lr	RD lr	Temperature τ	Weight α
FewRel	LLaMA-2-7B	5e-5	3e-6	4	0.5
	Qwen2.5-14B	5e-5	3e-6	4	0.2
TACRED	LLaMA-2-7B	1e-4	8e-5	4	1
	Qwen2.5-14B	7e-5	6e-6	2	0.9
FewRel-LT	LLaMA-2-7B	5e-5	3e-6	4	0.5
	Qwen2.5-14B	6e-5	3e-6	4	0.1

Table 2: Hyper-parameter settings.

prediction. As shown in line 1 of each dataset in Table 3, this results in a significant performance drop on three datasets. This indicates that the self-correcting inference strategy effectively coordinates RD and RP, enabling more accurate new relation prediction.

(2) *w/o. distillation strategy* and *w/o. relation predictor.* In our framework, RP is used not only as a teacher model for knowledge distillation during training but also for relation denoising and prediction during inference. To evaluate its impact, we design two ablated variants: one without RP during training, and another without RP in both training and inference. As shown in lines 2 and 3 of each dataset, removing RP during training causes a performance drop, which becomes more pronounced when RP is also removed during inference. These results demonstrate that the distillation strategy enhances RD's ability to discover new relations, and that RP plays a critical role in the overall effectiveness of our framework.

(3) *w/o. relation discoverer.* This variant uses RP directly to predict new relations based on demonstrations of known relation instances. However, Line 4 in each dataset shows that removing RD causes a notable performance drop across all datasets. An intuitive reason is that RP struggles to identify new relations when relying solely on known relation demonstrations. These results highlight the crucial role of RD in ensuring the practicality and effectiveness of our framework.

(4) *w/o. relation denoising stage.* Here, we remove the relation denoising stage from our framework and directly use the new relation instances discovered by RD as demonstrations for RP's relation prediction. This results in a significant performance drop across all three datasets (see line 5 of each dataset). This demonstrates that the denoising stage effectively selects noisy instances of new relations, thereby enabling RP to better understand new relations and accurately predict the relation of

Dataset	Method	B^3			V-measure			ARI	Classification		
		Prec.	Rec.	F_1	Hom.	Comp.	F_1		Prec.	Rec.	F_1
FewRel	**Ours**	0.647	0.700	**0.672**	0.790	0.809	**0.800**	**0.637**	0.750	0.737	**0.718**
	1 *w/o.* self-correcting inference strategy	0.657	0.571	0.611	0.801	0.724	0.761	0.570	0.753	0.619	0.624
	2 *w/o.* distillation strategy	0.625	0.677	0.650	0.770	0.789	0.780	0.613	0.746	0.726	0.708
	3 *w/o.* relation predictor	0.639	0.568	0.601	0.790	0.722	0.754	0.559	0.739	0.595	0.587
	4 *w/o.* relation discoverer	0.288	0.111	0.160	0.404	0.348	0.374	0.102	0.638	0.161	0.235
	5 *w/o.* relation denoising stage	0.582	0.627	0.604	0.745	0.762	0.754	0.570	0.710	0.691	0.674
	6 *w/o.* relation prediction stage	0.615	0.675	0.644	0.765	0.783	0.774	0.619	0.726	0.722	0.696
TACRED	**Ours**	0.739	0.700	**0.719**	0.769	0.731	**0.749**	**0.798**	0.665	0.742	**0.633**
	1 *w/o.* self-correcting inference strategy	0.741	0.681	0.710	0.770	0.676	0.720	0.699	0.563	0.666	0.541
	2 *w/o.* distillation strategy	0.725	0.680	0.702	0.752	0.707	0.729	0.773	0.650	0.735	0.625
	3 *w/o.* relation predictor	0.747	0.539	0.626	0.774	0.589	0.669	0.526	0.659	0.607	0.513
	4 *w/o.* relation discoverer	0.384	0.119	0.181	0.361	0.246	0.293	0.071	0.619	0.247	0.288
	5 *w/o.* relation denoising stage	0.710	0.658	0.683	0.747	0.703	0.724	0.726	0.645	0.717	0.605
	6 *w/o.* relation prediction stage	0.697	0.710	0.703	0.723	0.718	0.720	0.780	0.589	0.634	0.538
FewRel-LT	**Ours**	0.651	0.655	**0.653**	0.778	0.767	**0.773**	0.594	0.713	0.736	**0.698**
	1 *w/o.* self-correcting inference strategy	0.725	0.560	0.632	0.826	0.704	0.760	**0.615**	0.750	0.629	0.629
	2 *w/o.* distillation strategy	0.624	0.623	0.623	0.753	0.738	0.745	0.576	0.699	0.733	0.692
	3 *w/o.* relation predictor	0.700	0.553	0.618	0.809	0.697	0.749	0.572	0.704	0.606	0.579
	4 *w/o.* relation discoverer	0.341	0.120	0.177	0.444	0.351	0.392	0.108	0.573	0.169	0.241
	5 *w/o.* relation denoising stage	0.586	0.579	0.582	0.733	0.715	0.724	0.540	0.648	0.676	0.634
	6 *w/o.* relation prediction stage	0.591	0.643	0.616	0.736	0.742	0.739	0.581	0.692	0.726	0.676

Table 3: Ablation results on three RE datasets.

test instances.

(5) *w/o. relation prediction stage.* To assess the necessity of the relation prediction stage, this variant lets RD generate multiple candidate relations per test instance, with RP predicting the final relation from the candidate set. As shown in line 6 of each dataset, this approach consistently reduces final prediction performance, especially on TACRED. These results confirm that the relation prediction stage is essential for accurate new relation prediction.

In **Appendix** B, we further analyze the performance of relation discovery and the effect of the distillation loss weight α.

6 Related Work

Conventional RE methods (Song et al., 2019; Wu et al., 2022; Zhang et al., 2022, 2023b; Yue et al., 2024; Zhang et al., 2023a,c, 2025, 2024) cannot handle the continual emergence of new relations in real-world scenarios, which motivates the development of OpenRE. Previous approaches can be divided into two categories: Tagging-based (Yates et al., 2007; Etzioni et al., 2008) and Clustering-based (Yao et al., 2011; Marcheggiani and Titov, 2016; Elsahar et al., 2017; Wang et al., 2023). Tagging-based methods extract relations by analyzing the syntactic structure of sentences, but they often overlook semantic information, making clustering-based approaches more appealing. The clustering-based approaches aim to aggregate semantically related relation instances into the same cluster, with each cluster representing a potential new relation. Wu et al. (2019) leverages labeled data from predefined relations to train a model that can measure semantic similarity between relation instances. The learned similarity metric was then applied to cluster new relation instances. With the rise of pretrained language models (e.g., BERT (Devlin et al., 2019)), many studies (Hu et al., 2020; Zhao et al., 2021) have leveraged these models to encode an instance for obtaining its relational representation, as they are capable of capturing deep semantic information from text. Clustering is then performed on these representations to group semantically similar relation instances. However, the clustering-based approaches above cannot align clusters with specific relation types, restricting their applicability in downstream tasks. A recent study by Zhao et al. (2023) alleviates this issue by incorporating active learning into OpenRE, which actively selects representative instances for human annotation during the clustering process to align clusters with a specific relation type, but the need for human intervention severely restricts its practicality in real-world scenarios. Although a recent study by Wang et al. (2024) utilizes an API-based LLM as a phrase extractor to generate relational phrases for new relation instances, the use of a closed-source LLM incurs substantial API costs and cannot fully explore the RE ability of LLMs through training, making it difficult to distinguish fine-grained semantic relations. In this paper, we propose a novel OpenRE framework based on an open-source LLM to automatically discover new relations in real-world scenarios.

7 Conclusion

In this paper, we propose an LLM-based OpenRE framework, which aims to leverage the strong language understanding and generation abilities of LLMs to directly predict new relations for test instances without human intervention. The framework comprises two main components: (1) a Relation Discoverer (RD) that predicts new relations for test instances based on demonstrations built from training instances with known relations; and (2) a Relation Predictor (RP) that identifies the most likely relation for a test instance from n candidates, guided by demonstrations formed by their instances. To improve our framework's ability to predict new relations, we introduce a self-correcting inference strategy comprising three stages: relation discovery, relation denoising, and relation prediction. Specifically, we first use RD to preliminarily predict new relations for all test instances. Then, we apply RP to select high-reliability test instances for each new relation from the prediction results of RD. Finally, we employ RP to re-predict the relations of all test instances based on demonstrations constructed from these reliable test instances. Experimental results and in-depth analysis on three public datasets demonstrate the effectiveness of our framework.

Limitations

Despite its effectiveness, LLM-OREF has several limitations. First, our framework uses a fixed number of demonstrations during inference. Future studies should consider treating the number of demonstrations as a dynamic variable to better adapt to more complex scenarios in real-world applications. Second, we assume that the training data for known relations is noise-free. However, potential label noise in known relations could negatively impact new relation discovery, which future work should aim to address.

Acknowledgements

The project was supported by the National Natural Science Foundation of China (No. 62276219), Natural Science Foundation of Fujian Province of China (No.2024J011001), and the Public Technology Service Platform Project of Xiamen (No.3502Z20231043). We also thank the reviewers for their insightful comments.

References

Amit Bagga and Breck Baldwin. 1998. Algorithms for scoring coreference chains. In *The first international conference on language resources and evaluation workshop on linguistics coreference*, volume 1, pages 563–566. Citeseer.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.

Hady Elsahar, Elena Demidova, Simon Gottschalk, Christophe Gravier, and Frederique Laforest. 2017. Unsupervised open relation extraction. In *The Semantic Web: ESWC 2017 Satellite Events: ESWC 2017 Satellite Events, Portorož, Slovenia, May 28–June 1, 2017, Revised Selected Papers 14*, pages 12–16. Springer.

Oren Etzioni, Michele Banko, Stephen Soderland, and Daniel S Weld. 2008. Open information extraction from the web. *Communications of the ACM*, 51(12):68–74.

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.

William Hogan, Jiacheng Li, and Jingbo Shang. 2023. Open-world semi-supervised generalized relation discovery aligned in a real-world setting. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14227–14242.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Xuming Hu, Lijie Wen, Yusong Xu, Chenwei Zhang, and Philip S Yu. 2020. Selfore: Self-supervised relational feature learning for open relation extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3673–3682.

Lawrence Hubert and Phipps Arabie. 1985. Comparing partitions. *Journal of classification*, 2:193–218.

Heng Ji and Ralph Grishman. 2011. Knowledge base population: Successful approaches and challenges. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 1148–1158.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pages 611–626.

Yufei Li, Yuan Wang, and Xiaotao Huang. 2006. A relation-based search engine in semantic web. *IEEE transactions on knowledge and data engineering*, 19(2):273–282.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Diego Marcheggiani and Ivan Titov. 2016. Discrete-state variational autoencoders for joint discovery and factorization of relations. *Transactions of the Association for Computational Linguistics*, 4:231–244.

Juri Opitz and Sebastian Burst. 2019. Macro f1 and macro f1. *arXiv preprint arXiv:1911.03347*.

Pawan Rajpoot and Ankur Parikh. 2023. Gpt-finre: In-context learning for financial relation extraction using large language models. In *Proceedings of the Sixth Workshop on Financial Technology and Natural Language Processing*, pages 42–45.

Andrew Rosenberg and Julia Hirschberg. 2007. V-measure: A conditional entropy-based external cluster evaluation measure. *Empirical Methods in Natural Language Processing,Empirical Methods in Natural Language Processing*.

Linfeng Song, Yue Zhang, Daniel Gildea, Mo Yu, Zhiguo Wang, and Jinsong Su. 2019. Leveraging dependency forest for neural medical relation extraction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 208–218.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

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.

Zhen Wan, Fei Cheng, Zhuoyuan Mao, Qianying Liu, Haiyue Song, Jiwei Li, and Sadao Kurohashi. 2023. Gpt-re: In-context learning for relation extraction using large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3534–3547.

Jiaxin Wang, Lingling Zhang, Wee Sun Lee, Yujie Zhong, Liwei Kang, and Jun Liu. 2024. When phrases meet probabilities: enabling open relation extraction with cooperating large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13130–13147.

Jiaxin Wang, Lingling Zhang, Jun Liu, Xi Liang, Yujie Zhong, and Yaqiang Wu. 2022. Matchprompt: Prompt-based open relation extraction with semantic consistency guided clustering. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7875–7888.

Qing Wang, Yuepei Li, Qiao Qiao, Kang Zhou, and Qi Li. 2025. Towards a more generalized approach in open relation extraction. Proc. of 63rd Annual Meeting of the Association for Computational Linguistics.

Qing Wang, Kang Zhou, Qiao Qiao, Yuepei Li, and Qi Li. 2023. Improving unsupervised relation extraction by augmenting diverse sentence pairs. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12136–12147.

Changxing Wu, Liuwen Cao, Yubin Ge, Yang Liu, Min Zhang, and Jinsong Su. 2022. A label dependence-aware sequence generation model for multi-level implicit discourse relation recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11486–11494.

Ruidong Wu, Yuan Yao, Xu Han, Ruobing Xie, Zhiyuan Liu, Fen Lin, Leyu Lin, and Maosong Sun. 2019. Open relation extraction: Relational knowledge transfer from supervised data to unsupervised data. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 219–228.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Limin Yao, Aria Haghighi, Sebastian Riedel, and Andrew McCallum. 2011. Structured relation discovery using generative models. In *proceedings of the 2011 conference on empirical methods in natural language processing*, pages 1456–1466.

Alexander Yates, Michele Banko, Matthew Broadhead, Michael J Cafarella, Oren Etzioni, and Stephen Soderland. 2007. Textrunner: open information extraction on the web. In *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, pages 25–26.

Mo Yu, Wenpeng Yin, Kazi Saidul Hasan, Cicero dos Santos, Bing Xiang, and Bowen Zhou. 2017. Improved neural relation detection for knowledge base question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 571–581.

Hao Yue, Shaopeng Lai, Chengyi Yang, Liang Zhang, Junfeng Yao, and Jinsong Su. 2024. Towards better graph-based cross-document relation extraction via non-bridge entity enhancement and prediction debiasing. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 680–691.

Liang Zhang, Zijun Min, Jinsong Su, Pei Yu, Ante Wang, and Yidong Chen. 2023a. Exploring effective inter-encoder semantic interaction for document-level relation extraction. In *IJCAI*, pages 5278–5286.

Liang Zhang, Jinsong Su, Yidong Chen, Zhongjian Miao, Min Zijun, Qingguo Hu, and Xiaodong Shi. 2022. Towards better document-level relation extraction via iterative inference. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8306–8317.

Liang Zhang, Jinsong Su, Zijun Min, Zhongjian Miao, Qingguo Hu, Biao Fu, Xiaodong Shi, and Yidong Chen. 2023b. Exploring self-distillation based relational reasoning training for document-level relation extraction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 13967–13975.

Liang Zhang, Zhen Yang, Biao Fu, Ziyao Lu, Liangying Shao, Shiyu Liu, Fandong Meng, Jie Zhou, Xiaoli Wang, and Jinsong Su. 2024. Multi-level cross-modal alignment for speech relation extraction. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11975–11986.

Liang Zhang, Yang Zhang, Ziyao Lu, Fandong Meng, Jie Zhou, and Jinsong Su. 2025. A self-denoising model for robust few-shot relation extraction. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 26782–26797.

Liang Zhang, Chulun Zhou, Fandong Meng, Jinsong Su, Yidong Chen, and Jie Zhou. 2023c. Hypernetwork-based decoupling to improve model generalization for few-shot relation extraction. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 6213–6223.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Conference on empirical methods in natural language processing*.

Jun Zhao, Tao Gui, Qi Zhang, and Yaqian Zhou. 2021. A relation-oriented clustering method for open relation extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9707–9718.

Jun Zhao, Yongxin Zhang, Qi Zhang, Tao Gui, Zhongyu Wei, Minlong Peng, and Mingming Sun. 2023. Actively supervised clustering for open relation extraction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4985–4997.

A Details of Experiment Setup

A.1 Datasets

FewRel. It consists of 80 relation types, with 700 instances per relation. The first 40 relations are categorized into the known relation set, while the remaining 40 are categorized into the new relation set.

TACRED. It covers 41 relation types, in which the first 20 relations are categorized into the known relation set, while the remaining 21 are categorized into the new relation set.

FewRel-LT. Since FewRel is a class-balanced dataset that fails to reflect the long-tail distribution of relations in real-world scenarios, we follow Zhao et al. (2023) to construct the FewRel-LongTail (FewRel-LT) dataset. It shares the same split of known and new relations as FewRel, with each known relation keeping 700 instances. However, the number of instances for each new relation is adjusted to $n = \frac{700}{0.5*id+1}$, where id is from 0 to 39, representing each new relation.

A.2 Baselines

We compare our LLM-OREF with these OpenRE baselines: 1) **RW-HAC** (Elsahar et al., 2017) proposes an unsupervised method for OpenRE by reweighting word embeddings based on dependency paths and clustering the resulting relation representations. 2) **SelfORE** (Hu et al., 2020) proposes a self-supervised framework that iteratively clusters contextualized entity pair representations using adaptive soft clustering, and refines them through a relation classification module trained with pseudo labels. 3) **RSN** (Wu et al., 2019) learns similarity metrics of relations from labeled data of predefined relations, and then transfers the relational knowledge to identify new relations in unlabeled data. 4) **RoCORE** (Zhao et al., 2021) leverages the labeled data of predefined relations to learn a relation-oriented representation, while jointly optimizing objectives on both labeled and unlabeled data to improve new relation discovery. 5) **ASCORE** (Zhao et al., 2023) proposes an actively supervised clustering method for OpenRE, where clustering learning and human labeling are alternately performed, and an active labeling strategy is designed to select representative instances for labeling while dynamically discovering new relational clusters.

A.3 Prompt Template

Specifically, the prompt format used for both the Relation Discoverer and Relation Predictor is as follows:

> Prompt Template
>
> You are an expert in relationship extraction. Consider the following relationships to extract the relationship between the head entity and the tail entity from the text.
> The relationship must be in these possible relationships: [Relation Names].
> **Demonstrations:**
> **text:** [Text of Demo]
> **head_entity:** [Head entity of Demo]
> **tail_entity:** [Tail entity of Demo]
> **relationship:** [Relationship of Demo]
> . . .
> **text:** [Text of test instance]
> **head_entity:** [Head entity of test instance]
> **tail_entity:** [Tail entity of test instance]
> **relationship:**

B Analysis

B.1 The Performance of Relation Discovery

Dataset	Method	Precision	Recall	Macro-F_1	Accuracy	Pass@K
FewRel	LLM-OREF	0.750	0.737	0.718	0.737	-
	RD(K=1)	0.753	0.619	0.624	0.619	0.619
	RD(K=3)	0.501	0.401	0.400	0.396	0.788
	RD(K=5)	0.474	0.386	0.388	0.392	0.843
TACRED	LLM-OREF	0.665	0.742	0.633	0.757	-
	RD(K=1)	0.563	0.666	0.541	0.625	0.625
	RD(K=3)	0.425	0.395	0.349	0.431	0.791
	RD(K=5)	0.332	0.300	0.258	0.271	0.842
FewRel-LT	LLM-OREF	0.713	0.736	0.698	0.712	-
	RD(K=1)	0.750	0.629	0.629	0.607	0.607
	RD(K=3)	0.441	0.403	0.380	0.392	0.768
	RD(K=5)	0.448	0.406	0.384	0.423	0.833

Table 4: Performance of Relation Discovery under different numbers of predictions K.

In the relation discovery stage, we make multiple predictions for each test instance to obtain multiple prediction relations for improving the recall of the Relation Discoverer (RD) in discovering new relations. As shown in Table 4, the Pass@K metric significantly improves with increasing number of predictions K, indicating that the relations of test instances are increasingly recalled correctly by the RD. Such improved recall is critical for enabling more effective relation denoising in the subsequent stage. Therefore, we set the number of predictions $K = 3$ across all datasets. Furthermore, our framework consistently outperforms the RD on all evaluation metrics, further demonstrating the effectiveness of our approach.

B.2 The Effect of Distillation Loss Weight α

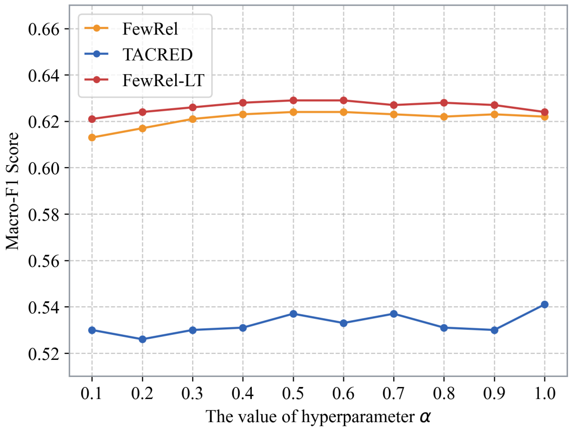

Figure 5: Performance with different weight α of $\mathcal{L}_{\text{KD}}$ for Relation Discoverer.

To investigate the impact of the distillation loss weight α on the ability of the RD to discover new relations, experiments are conducted to compare the performance of the RD by varying the value of α. Figure 5 shows that the RD achieves the best performance on the FewRel and FewRel-LT when $\alpha = 0.5$, and on the TACRED when $\alpha = 1$. Meanwhile, we observe that the distillation strategy exhibits low sensitivity to the value of α. Therefore, we set $\alpha = 0.5$ for the FewRel and FewRel-LT, and $\alpha = 1$ for the TACRED.

C Discussion

C.1 OpenRE Setting

Recent studies (Hogan et al., 2023; Wang et al., 2025) suggest that the unlabeled dataset is typically a mixture of known and new relations. For a fair comparison, we adopt the test setting used by most existing works, where the test set contains only instances of new relations. This commonly used setting is generally more challenging than

Dataset	**Method**	**F$_1$**	**B^3-F$_1$**	**V-measure F$_1$**	**ARI**
FewRel	KNoRD (Hogan et al., 2023)	0.774	0.732	0.730	0.695
	MixORE (Wang et al., 2025)	0.833	0.897	0.880	0.882
	Ours	**0.941**	**0.898**	**0.902**	**0.883**
TACRED	KNoRD (Hogan et al., 2023)	0.852	0.768	0.788	0.719
	MixORE (Wang et al., 2025)	0.883	0.868	0.860	0.847
	Ours	**0.907**	**0.868**	**0.867**	**0.871**
FewRel-LT	KNoRD (Hogan et al., 2023)	0.867	0.639	0.731	0.509
	MixORE (Wang et al., 2025)	0.916	0.875	0.861	0.893
	Ours	**0.959**	**0.890**	**0.896**	**0.898**

Table 5: Performance of Relation Discovery under different numbers of predictions K.

the mixed setting, as RE models typically perform better on relations they have encountered during training. Consequently, the performance of RE models under this widely used setting can be regarded as a lower bound of their performance in the mixed setting. Here, we further compare our method with the relevant baselines under the mixed setting. From Table 5, we observed that our method still outperforms all baselines under this setting, further validating the effectiveness and robustness of our approach.