Prompt Optimization for Relation Extraction using Reinforcement Learning

Ying Liu[1], Shuai Dong[1], Zibo Cui[1], TengQi Ye[2], Gang Wu[3,*]
[1]Software College, Northeastern University, China
[2]Meta, USA
[3]School of Computer Science and Engineering, Northeastern University, China

Abstract

Relation extraction is a fundamental task in information extraction. Still, existing supervised approaches rely heavily on large-scale annotated data, limiting their applicability in domain-specific and low-resource scenarios. Prompt-based methods with large language models provide a parameter-efficient alternative; however, their performance is susceptible to prompt design, which often requires extensive domain expertise and heuristic trial-and-error. We propose REPO, a reinforcement learning-based automated prompt optimization framework for domain relation extraction. REPO formulates prompt construction as a structured, sequential decision-making problem, optimizing prompt quality through interaction with a black-box LLM. To enable efficient and stable optimization, we introduce a two-stage framework comprising an initial prompt-construction stage that generates semantically grounded candidates and a DRL-based refinement stage that iteratively improves prompts within a constrained, domain-aware action space. We further design a composite evaluation metric that integrates extraction accuracy and semantic consistency to serve as a dense reward signal. Extensive experiments on multiple relation extraction datasets across medical, financial, legal, and news domains demonstrate that REPO consistently outperforms existing prompt-based methods and supervised baselines. Ablation studies further confirm the effectiveness and robustness of the proposed DRL-based prompt optimization strategy. Our code is available at https://github.com/dddong2-star/REPO.

1 Introduction

Relation extraction (RE) as a core task in information extraction plays a critical role in applications such as knowledge graph construction and alignment (Chen et al., 2022; Zhang et al., 2022b), question answering (Luo et al., 2018), and knowledge retrieval (Yang, 2020). Deep learning approaches, particularly supervised learning methods, have significantly improved RE performance. However, these methods rely heavily on large-scale, high-quality annotated datasets (Zeng et al., 2014), severely limiting their scalability across domains, especially in specialized fields such as medicine, law, and finance.

The in-context prompting paradigm of large language models (LLMs), which uses natural-language instructions to enable few-shot learning, provides a direct mechanism to reduce this dependency on annotated data. Encouraged by these advances, researchers have begun exploring in-context prompts for RE tasks, achieving encouraging results. Liu et al. (2024) proposes the Self-Prompting framework, which includes synonym and label generation and sentence rewriting to optimize the prompt. Lu et al. (2022) proposed the SURE model to guide the model's objectives by applying entity marking and sentence rewriting strategies. These approaches primarily rely on manually constructed templates, fully utilizing prior knowledge embedded in PLMs (Pre-trained Language Models). Li et al. (2023a) summarizes the semantic relation between head and tail entities to reason multi-step for summarization, and question and answer. While this method reduces task complexity, its label mapping process remains relatively intricate, leaving room for further optimization. More importantly, the manual design of high-quality prompts itself is a labor-intensive process that demands extensive domain expertise and iterative experimentation (Jiang et al., 2020), which fundamentally limits the efficiency and broader applicability of these methods.

Against this background, automated prompt generation have emerged as a key approach, aiming to reduce manual overhead and improve scal-

*Corresponding author: wugang@mail.neu.edu.cn

Findings of the Association for Computational Linguistics: ACL 2026, pages 42321–42334

ability. Luo et al. (2025) introduces TAPO, a multitask-aware prompt optimization framework integrating task-aware metric selection, multi-metric evaluation, and evolution-based prompt refinement which improves prompt generation and enhances LLM adaptability across diverse tasks. As LLMs are highly sensitive to the surface form of prompts. Slight variations in wording, structure, or example ordering can lead to substantial performance fluctuations, especially in structured prediction tasks. Zhao et al. (2024) propose a method that includes automatic template generation, weighting, grouping, and optimization, effectively addressing template bias. Nevertheless, it requires additional entity-type annotations that lack generalization.

However, the effectiveness of automated prompt generation for relation extraction is constrained by two key challenges. On the one hand, prompt optimization for relation extraction entails a combinatorial and sequential decision process over heterogeneous components, including template structures, lexical realizations, entity descriptions, relation constraints, and example formats. Reliance on unstructured search strategies makes it difficult to efficiently explore this space, leading to slow convergence and high computational cost. On the other hand, existing automated prompts lack explicit modeling of domain-specific relation semantics, which limits their generalization to domain-specific relation extraction scenarios such as medicine and law.

To address these challenges, we propose a structured, controllable two-stage framework for automated prompt optimization in domain relation extraction, employing deep reinforcement learning (DRL) for fine-grained prompt refinement. During the initial prompt construction stage, we leverage large language models' semantic understanding to generate candidate prompts from example instances automatically. In the prompt-optimization stage, DRL is introduced to learn prompt-enhancement strategies that iteratively refine these candidates toward optimal prompts. Furthermore, by analyzing the relation extraction task, we propose a comprehensive prompt quality evaluation metric that integrates supervised metrics with similarity-based metrics, enabling a multidimensional assessment of prompt effectiveness.

Our main contributions are as follows.

1. **Deep Learning based Prompt Optimization Framework**. To our best knowledge, we are the first to propose a reinforcement learning framework for the prompt optimization in relation extraction.

2. **Cost Effective Approach**. In most public datasets, our approach consumes less than 3M tokens with less than 5K API calls.

3. **Robust Performance**. Our extensive experiments demonstrates that REPO consistently outperforms existing supervised baselines and prompt-based counterparts.

2 Related Works

Relation extraction (RE) aims to identify structured triples ⟨head entity, relation, tail entity⟩ from natural language text. Traditional deep learning approaches to RE can be broadly categorized into pipeline-based methods (Xu et al., 2015), span-based methods (Dixit and Al-Onaizan, 2019; Mandya et al., 2020), sequence-to-sequence (Seq2Seq) methods (Nayak and Ng, 2020; Zeng et al., 2020), and machine reading comprehension (MRC)-based methods (Li et al., 2019; Zhao et al., 2021). With the emergence of large language models (LLMs), recent work has explored leveraging their strong language understanding and reasoning capabilities for relation extraction.

One line of research enhances RE by integrating LLMs with auxiliary modules or external reasoning frameworks. For example, Li et al. (2023b) combine LLMs with a natural language inference module to generate relational triplets, achieving improved performance on document-level RE tasks. While effective, these approaches typically require additional model components and customized training pipelines, increasing architectural complexity and computational overhead and limiting scalability.

Another line of work focuses on fine-tuning LLMs for relation extraction. Wadhwa et al. (2023) train the Flan-T5 model using Chain-of-Thought (CoT) style explanations to enhance relational reasoning and extraction accuracy. Despite their strong empirical performance, fine-tuning-based methods generally rely on large-scale, high-quality annotated datasets and incur substantial computational costs, making them less practical in low-resource or domain-specific scenarios.

More recently, prompt-based relation extraction has emerged as a parameter-efficient alterna-

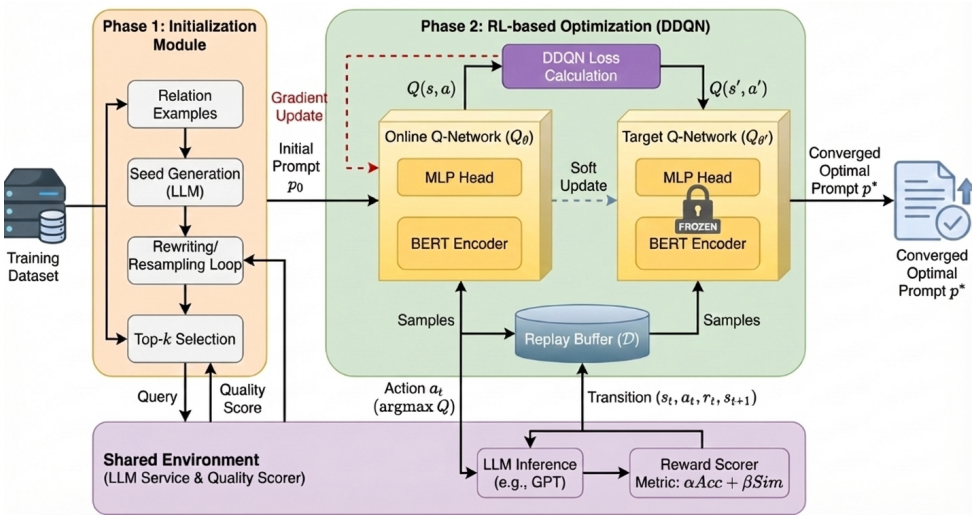

Figure 1: An overview of our framework.

tive, in which LLMs perform RE via carefully designed prompts without updating model parameters (Brown et al., 2020). Wan et al. (2023) improve entity–relation alignment by retrieving task-relevant examples and incorporating logical reasoning demonstrations into prompts. Gutierrez et al. (2022) introduce a k-nearest neighbors (kNN) retrieval module to select representative in-context examples and systematically construct effective prompts, improving GPT-3s performance on bioinformatics information extraction tasks. Zhao et al. (2024) further explore prompt template design by varying the number of demonstrations and relation types, achieving notable improvements in entityrelation extraction.

Although prompt-based methods substantially reduce annotation requirements and deployment costs, they still face several fundamental challenges. Prior studies have shown that LLMs are highly sensitive to prompt formulations, and even semantically similar prompts can yield significantly different performance. Moreover, existing prompt construction strategies are largely heuristic, lacking unified optimization objectives or principled search mechanisms. Most approaches rely heavily on manual design choices, domain expertise, and extensive trial-and-error, particularly in domain-specific RE tasks. As a result, prompt quality is difficult to control, performance stability is hard to guarantee, and adapting prompts across domains or datasets remains costly and inefficient.

In contrast to prior prompt-based RE methods that treat prompt design as a static or heuristic process, our work formulates prompt construction as a structured and sequential decision-making problem. By explicitly incorporating domain knowledge into a constrained action space and optimizing promptsvia a two-stage reinforcement learning framework, our approach provides a principled solution for robust prompt optimization in relation extraction.

3 Method

To address the aforementioned challenges, we apply Reinforcement Learning to Relation Extraction using Prompt Optimization (REPO). We formulate automated prompt construction for relation extraction as a structured optimization problem, consisting of two sequential stages: (i) an initial prompt generation stage for producing semantically reasonable seed prompts, and (ii) a reinforcement-learning-based prompt optimization stage for controlled and performance-driven refinement. We also propose a composite metric termed *Relation Extraction Prompt Quality Score* (REPQS). REPQS measures prompt quality from both prediction accuracy and semantic alignment perspectives, providing a more robust signal than conventional task metrics alone. An overview of the framework is illustrated in Figure 1.

3.1 Reward Signal and Evaluation Metric

To ensure that prompt optimization aligns with downstream task objectives, we design a comprehensive evaluation metric, REPQS, that serves as both an evaluation criterion and a reinforcement learning reward signal. This metric jointly considers extraction accuracy and semantic consistency,

enabling more precise and stable optimization of prompts for relation extraction.

Given a prompt p and an input text x, LLM output is $y = \text{LLM}(p \mid x)$, and we use y^* as the ground-truth relational triple. The REPQS score is defined as:

$$\text{REPQS}(p) = \alpha \cdot F_1(y, y^*) + \beta \cdot Sim(y, y^*) \quad (1)$$

α and β are weighting coefficients that balance task accuracy and semantic consistency.

F1-based Evaluation. For the $F_1(\cdot)$ component, relation triples $\langle e_1, r, e_2 \rangle$ are extracted from the LLM output and compared against annotated ground-truth triples. Precision and recall are computed based on exact matching of entities and relations.

Semantic Similarity Evaluation To capture partial correctness and semantic proximity, the $Sim(\cdot)$ component measures similarity at the embedding level. Specifically, we employ the SBERT (Reimers and Gurevych, 2019) to obtain vector representations for entities and relations in both predicted and ground-truth triples. For each predicted entity (or relation), cosine similarities with all ground-truth entities (or relations) are computed, and the maximum similarity score is selected. The final similarity score is obtained by averaging the entity-level and relation-level similarities and normalizing the result to $[0, 1]$.

By combining exact matching and semantic similarity, REPQS provides a dense, informative reward signal, enabling stable prompt optimization even under limited supervision and mitigating the sparsity issues associated with pure accuracy-based rewards.

3.2 Stage 1: Prompt Initialization

To provide a stable, semantically grounded initialization for subsequent reinforcement learning, we design a heuristic algorithm (Algorithm 1) to construct an initial prompt set automatically. This stage aims to generate a compact, diverse, and high-quality search space rather than exhaustively exploring free-form prompt variations.

The algorithm first constructs the initial seed set p_{seed} by selecting the Top-k% prompts via REPQS (Line 4). It then enters an iterative optimization loop (Lines 5 to 21) until the maximum number of rounds, max_round, is reached or convergence is achieved. In each round, a LLM rewrites the current seed prompts to generate semantically equivalent variants with more optimal structures, which are subsequently merged with the original seed set. Based on REPQS, the Top-k% prompts are filtered to maintain quality and reduce the search space, and the optimal prompt $best\_prompt$ is updated.

Algorithm 1 Prompt initialization

Require: Supply relation extraction dataset R, number of iterations max_round, initial prompt p

1: previous_score $\leftarrow$ 0, best_score $\leftarrow$ -1, best_prompt $\leftarrow$ None
2: $r \leftarrow$ random_sample(R)
3: $I \leftarrow \text{LLM}(r)$
4: $p_{\text{seed}} \leftarrow \{p_i \mid p_i \in I, \text{Rank}(\text{REPQS}(p_i)) \leq \text{len}(I) \times k\%\}$
5: **while** round $<$ max_round **do**
6: $I \leftarrow$ re_write(p_{seed})
7: $I \leftarrow p_{\text{seed}} \cup I$
8: update_best_prompt(I)
9: $I \leftarrow \{p_i \mid p_i \in I, \text{Rank}(\text{REPQS}(p_i)) \leq \text{len}(I) \times k\%\}$
10: **if** round $\bmod 5 = 0$ **then**
11: resample(I)
12: **end if**
13: current_score $\leftarrow$ mean $(\sum \text{REPQS}(p_i))$, $p_i \in I$
14: delta $\leftarrow$ current_score $-$ previous_score
15: **if** abs(delta) $<$ 0.05 **and** holds for three consecutive iterations **then**
16: **break**
17: **end if**
18: $p_{\text{seed}} \leftarrow I$
19: previous_score $\leftarrow$ current_score
20: round $\leftarrow$ round $+ 1$
21: **end while**
22: **return** best_prompt

To preserve semantic diversity and avoid premature convergence, the algorithm resamples the filtered prompt set every five rounds. Convergence is determined by calculating the difference δ between the average REPQS scores of the current and previous rounds. If the absolute value of delta is less than 0.05 and this condition holds for three consecutive rounds, the algorithm terminates early. During the iteration, the seed set p_{seed} and the historical average score are continuously updated.

Upon completing the loop, the algorithm outputs the optimal prompt $best_{prompt}$ with the high-

est score throughout the optimization process. This stage ultimately yields a compact, diverse, and task-aligned prompt set, significantly reducing the search space and laying a solid foundation for stable, efficient reinforcement learning-based optimization in the subsequent stage.

3.3 Stage 2: Prompt Optimization using Reinforcement Learning

With the seed prompts obtained in Stage 1, we further refine prompts using reinforcement learning for domain-specific RE tasks. Namely, we consider prompt optimization as a sequential decision-making problem over a structured, interpretable action space, enabling controlled exploration while preserving semantic validity.

MDP formulation. Prompt optimization is formulated as a Markov Decision Process (MDP) defined by (S, A, R). Since the transition dynamics are unknown and the LLM operates as a black box, a model-free reinforcement learning approach is adopted.

The state $s_t \in S$ represents the semantic state of the current prompt at step t. Concretely, the prompt text is encoded by a BERT encoder, and the hidden representations from the final layer are aggregated as a contextualized semantic embedding. This representation jointly captures prompt structure, entity-related cues, and relation-oriented semantics, which are critical for guiding effective prompt edits.

Action space. The action space in REPO is constrained to a set of task-relevant prompt editing operations, such as refining relation descriptions and adjusting entity role specifications. This design ensures semantic validity and stable optimization trajectories. In contrast to arbitrary operations, the pre-defined action space reduces semantic drift and enables more reliable exploration during RL-based prompt optimization. The action space A consists of 11 predefined prompt-editing operations from 7 groups, which are detailed in Table 1), designed according to both relation extraction characteristics and domain-specific prior knowledge.

By constraining the optimization process to these semantically meaningful operations, the action space balances expressiveness and tractability, while maintaining interpretability and reducing the risk of semantic drift commonly observed in automatic prompt generation.

Reward design. Given a prompt p and an input text x, the concatenated input $[p : x]$ is fed into a black-box LLM to produce output y. We adopt REPQS as the basic reward signal. However, due to variations in input difficulty and stochasticity in LLM inference, single-instance rewards can be unstable.

To address this issue, we define the reinforcement learning reward at the prompt level as the mean REPQS over a dataset $\mathcal{T}(x, y)$:

$$\text{M-REPQS}(p) = \frac{1}{n} \sum_{i=1}^{n} R_p(x_i), \tag{2}$$

where $R_p(x_i)$ denotes the REPQS score of prompt p on input x_i. This averaged reward reduces variance and provides a more reliable optimization signal for policy learning.

Optimization algorithm. We adopt the Double Deep Q-Network (DDQN) algorithm (Van Hasselt et al., 2016) to learn the optimal prompt-editing policy and alleviate Q-value overestimation. The BERT-encoded prompt representation is fed into a task-specific multilayer perceptron (MLP) to estimate action-value functions for all candidate actions.

To improve training stability and sample efficiency, an experience replay buffer stores transitions (s_t, a_t, r_t, s_{t+1}) collected during interaction with the environment. During training, mini-batches are randomly sampled from the buffer to break temporal correlations. Action selection follows an ϵ-greedy strategy with linear decay, encouraging exploration in early stages and exploitation of high-value actions in later stages. The online and target networks are periodically synchronized to further stabilize learning.

Through iterative interaction and reward-guided updates, the agent progressively learns to apply practical prompt-editing actions, yielding optimized prompts that demonstrate strong performance and robustness in low-resource relation extraction settings.

4 Experiments

4.1 Datasets

We evaluate our method on four relation extraction datasets from diverse domains. For each dataset, we select representative relation types to construct a focused evaluation setting.

Category	Purpose	Action
Original Prompt	Allows the agent to keep the current prompt unchanged to prevent over-optimization when a locally optimal prompt has been reached.	Avoid over-optimization
Structure	Modifies sentence patterns to improve linguistic diversity and robustness to varied expression styles.	Change sentence structure
Vocabulary	Aligns relation-related expressions with canonical label definitions, reducing semantic ambiguity across different surface forms.	Regularize and correct keywords
Entity Information	Strengthens the model's discrimination of entity roles by adding entity types, descriptive attributes, and positional emphasis.	Add entity type
		Add entity description
		Enhance entity position
Relation Information	Supplements relation semantics and explicitly clarifies relation directionality (head vs. tail entity).	Add relation description
		Enhance relation direction
Examples	Incorporates a small number of inputoutput pairs or sentence templates in a few-shot manner to provide task demonstrations.	Add input/output pairs
		Add sentence templates
Output Format	Enforces structured and label-consistent outputs.	Enhance output format

Table 1: Prompt optimization action space.

(1) **CMeIE** (Zhang et al., 2022a): It is a subset of CBLUE (Chinese Biomedical Language Understanding Evaluation) dataset, which is a Chinese biomedical information benchmark. We select samples with the relations *etiology*, *drug treatment*, and *clinical manifestation*. Finally, 864 instances remain.

(2) **FinCUGE** (Lu et al., 2023): It is a large-scale financial corpus with approximately 300GB of raw text from four different sources. We adopt samples annotated with the relations *cooperation* and *ownership* are retained, yielding 1,219 instances.

(3) **LexEval** (Li et al., 2024): A Chinese legal case dataset. We select four relation types: *drug trafficking*, *human trafficking*, *illegal harboring*, and *possession*, resulting in 497 instances.

(4) **LCN**: We collect the Listed Company News (LCN) from the public news. After data cleaning and manual annotation, we extract samples with the relations *supplier*, *production*, and *composition*, resulting in 1,953 instances. The dataset is accessible from our github repository.

4.2 Baseline Methods

We compare our method against prompt-based and supervised relation extraction approaches.

(1) **APE** (Zhou et al., 2022): an automatic prompt generation framework that produces multiple candidate prompts from examples and iteratively selects and rewrites prompts based on performance scores.

(2) **OPRO** (Yang et al., 2023):a prompt optimization approach that formulates prompt refinement as a natural language optimization task, where the large language model iteratively generates and evaluates new prompts.

(3) **SPO** (Xiang et al., 2025): a prompt optimization framework that improves prompts via self-supervised comparison of model outputs without relying on annotated ground-truth labels.

(4) **CasRel** (Wei et al., 2020): a supervised neural relation extraction model based on cascade and residual learning, which serves as a representative end-to-end baseline.

4.3 Experimental Settings

For each dataset, the ratio of training, validation, and test sets is 1:1:8. The training and validation sets are used for prompt construction and optimization, while the test set is reserved for final evaluation. We set the hyperparameter in REPO score as $\alpha = 5$, $\beta = 1$. Table 2 lists several key hyperparameters.

We report **Precision (P)**, **Recall (R)**, and **F1-score** as evaluation metrics. Precision measures the proportion of predicted relations that are correct, recall measures the proportion of gold relations that are successfully identified, and F1-score is the harmonic mean of precision and recall. All reported results are computed on the test sets.

To further explore the potential value of prompt optimization in improving model performance, we also fine tune REPO (REPO-FT) applied to the main large language model Qwen2.5-7B-Instruct-1M by using the LoRA fine-tuning framework(Hu et al., 2022), and compare the performance of the fine-tuned model with the GPT-4o model.

Table 2: Key Hyperparameter Settings

Module	Hyperparameter	Value
Initial Prompt Construction & Resampling	Max Iteration Rounds (`max_round`)	6
	Seed Prompt Selection Ratio ($k\%$)	50%
	Iteration Stopping Threshold (`threshold`)	0.01
Reinforcement Learning (DDQN)	Discount Factor (γ)	0.99
	Learning Rate (`learning_rate`)	1e-4
	Initial Exploration Rate ($\epsilon_{\max}$)	0.95
	Minimum Exploration Rate ($\epsilon_{\min}$)	0.05
	Exploration Decay Rate (`decay`)	0.995
REPQS Metric	Accuracy Weight (α)	5
	Similarity Weight (β)	1

4.4 Experimental Results and Analysis

Main Results Table 3 summarizes results on four relation extraction datasets. REPO consistently achieves the highest F1 scores, demonstrating effectiveness and robustness in low-resource settings. The performance gains are further supported by the ablation study in Table 4, which shows a consistent drop in F1 when the prompt optimization component is removed.

REPO attains F1 scores of 0.72 on LexEval, 0.60 on FinCUGE, 0.58 on CMeIE, and 0.56 on LCN, outperforming all prompt-based baselines: APE, OPRO, and SPO. Compared to these, REPO improves F1 by roughly $4\% - 8\%$, with the most significant gains on FinCUGE and LCN. OPRO and SPO sometimes yield higher recall but have consistently lower precision, resulting in lower F1 scores. REPO more evenly balances precision and recall across datasets, indicating that its structured reinforcement-learning prompt optimization reduces false negatives without adding noise.

Compared with the supervised baseline, CasRel and REPO demonstrate competitive or superior performance under limited-annotation conditions. CasRel achieves substantial precision on LexEval, where data is abundant, and relations are less diverse. Still, its performance degrades on FinCUGE, CMeIE, and LCN due to scarce training data and more varied relation expressions. In contrast, REPO, which does not rely on task-specific supervised training, exhibits more stable performance, highlighting its advantage when labeled data are expensive or difficult to obtain.

We evaluate model fine-tuning by comparing REPO to its fine-tuned variant, REPO-FT, on Qwen2.5-7B-Instruct-1M using LoRA. REPO-FT achieves F1 improvements of 0.08 on LexEval, 0.05 on FinCUGE, and 0.07 on CMeIE, with a slight decrease on LCN. On average, fine-tuning increases F1 by 4.7%, confirming that task-aware parameter adaptation enhances RL-based prompt optimization.

Across all four data sets spanning the medical, financial, legal, and news domains, REPO consistently improves performance, indicating strong cross-domain generalization. These results suggest that learning reusable, constrained prompt-editing strategies via RL-based prompt optimization enhances the stability and adaptability of prompt-based relation extraction methods.

Ablation Study To further analyze the sources of the performance gains observed in the main experiments, we conduct ablation studies to disentangle the contributions of the two key components in the REPO framework: the initial prompt construction stage and the reinforcement learning-based prompt optimization stage. Specifically, we compare the full REPO model with two ablated variants: (i) **Only Init**, which removes the RL-based optimization stage and retains only the initial prompt construction, and (ii) **Only RL**, which removes the initial prompt construction stage and performs prompt optimization solely through RL. All other components and experimental settings are kept identical.

Table 4 reports the F1 scores of the full model and the ablated variants across four datasets. The full REPO model consistently achieves the best performance on all datasets, outperforming both ablated variants. Compared to **Only Init** and **Only RL**, the full model yields absolute F1 improvements of 0.02/0.02 on LexEval, 0.04/0.07 on FinCUGE, 0.03/0.05 on CMeIE, and 0.04/0.05 on LCN, respectively. These results indicate that the two stages are complementary: the initial prompt construction provides a strong and stable starting point, while RL-based optimization further refines

Table 3: Experimental Results of Different Models on Four Datasets. **Bold** denotes the best results on each dataset.

Method	Model	Metric	Dataset			
			Lexeval	FinCUGE	CMeIE	LCN
APE	**GPT-4o**	P	0.52	0.47	0.33	0.42
		R	0.70	0.49	0.59	0.46
		F1	0.60	0.48	0.42	0.44
OPRO	**GPT-4o**	P	0.61	0.49	0.46	0.44
		R	0.84	0.60	0.69	0.74
		F1	0.71	0.54	0.55	0.51
SPO	**GPT-4o**	P	0.38	0.46	0.43	0.42
		R	0.74	0.78	0.70	0.50
		F1	0.50	0.57	0.53	0.46
CasRel	**BERT**	P	0.89	0.36	0.42	0.54
		R	0.50	0.55	0.48	0.23
		F1	0.64	0.44	0.45	0.32
REPO	**GPT-4o**	P	0.59	0.55	0.48	0.51
		R	0.93	0.75	0.72	0.62
		F1	**0.72**	**0.60**	**0.58**	**0.56**
REPO-FT	**Qwen**	P	0.71	0.59	0.59	0.40
		R	0.91	0.73	0.73	0.70
		F1	0.80	0.65	0.65	0.51

Table 4: Comparison of F1-Scores between Ablation Experiment and Full Model

Dataset	Only RL	Only Init	Full
LexEval	0.70	0.70	0.72
FinCUGE	0.56	0.53	0.60
CMeIE	0.55	0.53	0.58
LCN	0.52	0.51	0.56

prompts toward higher-quality solutions.

The performance gains vary across datasets, consistent with the main experimental findings. The most notable improvements are observed on FinCUGE, which contains diverse relation expressions and limited annotated data, indicating that RL-based prompt optimization is particularly effective in challenging low-resource settings. On LexEval, where baseline performance is relatively strong, the improvements are smaller but consistent, suggesting stable rather than dataset-specific gains. Overall, the ablation results demonstrate that the initial prompt construction stage effectively constrains the search space, while the RL-based optimization stage further refines prompts, and their combination enables robust relation extraction across different domains.

5 Conclusions

This paper proposes REPO to improve performance on relation extraction tasks. We have designed a two-stage framework that combines heuristic initialization with DRL optimization. This design significantly reduces the adequate search space, improves optimization efficiency, and mitigates the instability commonly observed in unconstrained prompt search. Compared with representative relation extraction methods, REPO outperforms the baselines across four datasets, fully demonstrating its strong robustness and generalization in low-resource and cross-domain scenarios. In addition, the LoRA-based fine-tuning experiment (REPO-FT) further verifies the framework's potential to enhance the task adaptability of large language models. Through ablation experiments, we also confirm that the RL-based prompt-optimization component significantly improves performance on the relation extraction task.

References

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.

Xiang Chen, Ningyu Zhang, Lei Li, Shumin Deng, Chuanqi Tan, Changliang Xu, Fei Huang, Luo Si, and Huajun Chen. 2022. Hybrid transformer with multi-level fusion for multimodal knowledge graph completion. In *Proceedings of the 45th interna-*

tional ACM SIGIR conference on research and development in information retrieval, pages 904–915.

Kalpit Dixit and Yaser Al-Onaizan. 2019. Span-level model for relation extraction. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 5308–5314.

Bernal Jimenez Gutierrez, Nikolas McNeal, Clayton Washington, You Chen, Lang Li, Huan Sun, and Yu Su. 2022. Thinking about gpt-3 in-context learning for biomedical ie? think again. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4497–4512.

Edward J. Hu, Yelong Shen, Phil Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Swabha Wang, Lu Wang, Weizhu Chen, and Denny Zhou. 2022. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*. Accepted to ICLR 2022.

Zhengbao Jiang, Frank F Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

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

Haitao Li, You Chen, Qingyao Ai, Yueyue Wu, Ruizhe Zhang, and Yiqun Liu. 2024. Lexeval: A comprehensive chinese legal benchmark for evaluating large language models. *Advances in Neural Information Processing Systems*, 37:25061–25094.

Junpeng Li, Zixia Jia, and Zilong Zheng. 2023b. Semi-automatic data enhancement for document-level relation extraction with distant supervision from large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5495–5505.

Xiaoya Li, Fan Yin, Zijun Sun, Xiayu Li, Arianna Yuan, Duo Chai, Mingxin Zhou, and Jiwei Li. 2019. Entity-relation extraction as multi-turn question answering. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 1340–1350.

Siyi Liu, Yang Li, Jiang Li, Shan Yang, and Yunshi Lan. 2024. Unleashing the power of large language models in zero-shot relation extraction via self-prompting. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 13147–13161.

Dakuan Lu, Hengkui Wu, Jiaqing Liang, Yipei Xu, Qianyu He, Yipeng Geng, Mengkun Han, Yingsi Xin, and Yanghua Xiao. 2023. Bbt-fin: Comprehensive construction of chinese financial domain pre-trained language model, corpus and benchmark. *arXiv preprint arXiv:2302.09432*.

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.

Kangqi Luo, Fengli Lin, Xusheng Luo, and Kenny Zhu. 2018. Knowledge base question answering via encoding of complex query graphs. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 2185–2194.

Wenxin Luo, Weirui Wang, Xiaopeng Li, Weibo Zhou, Pengyue Jia, and Xiangyu Zhao. 2025. Tapo: Task-referenced adaptation for prompt optimization. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Angrosh Mandya, Danushka Bollegala, and Frans Coenen. 2020. Graph convolution over multiple dependency sub-graphs for relation extraction. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6424–6435. International Committee on Computational Linguistics.

Tapas Nayak and Hwee Tou Ng. 2020. Effective modeling of encoder-decoder architecture for joint entity and relation extraction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8528–8535.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Hado Van Hasselt, Arthur Guez, and David Silver. 2016. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30.

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.

Zhepei Wei, Jianlin Su, Yue Wang, Yuan Tian, and Yi Chang. 2020. A novel cascade binary tagging framework for relational triple extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1476–1488.

Jinyu Xiang, Jiayi Zhang, Zhaoyang Yu, Xinbing Liang, Fengwei Teng, Jinhao Tu, Fashen Ren, Xiangru Tang, Sirui Hong, Chenglin Wu, and Yuyu Luo. 2025. Self-supervised prompt optimization. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 9017–9041.

Yan Xu, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng, and Zhi Jin. 2015. Classifying relations via long short term memory networks along shortest dependency paths. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1785–1794.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. 2023. Large language models as optimizers. In *The Twelfth International Conference on Learning Representations*.

Zuoxi Yang. 2020. Biomedical information retrieval incorporating knowledge graph for explainable precision medicine. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2486–2486.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In *Proceedings of COLING 2014, the 25th international conference on computational linguistics: technical papers*, pages 2335–2344.

Daojian Zeng, Haoran Zhang, and Qianying Liu. 2020. Copymtl: Copy mechanism for joint extraction of entities and relations with multi-task learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 9507–9514.

Ningyu Zhang, Mosha Chen, Zhen Bi, Xiaozhuan Liang, Lei Li, Xin Shang, Kangping Yin, Chuanqi Tan, Jian Xu, Fei Huang, and 1 others. 2022a. Cblue: A chinese biomedical language understanding evaluation benchmark. In *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 7888–7915.

Rui Zhang, Bayu Distiawan Trisedya, Miao Li, Yong Jiang, and Jianzhong Qi. 2022b. A benchmark and comprehensive survey on knowledge graph entity alignment via representation learning. *The VLDB Journal*, 31(5):1143–1168.

Tianyang Zhao, Zhao Yan, Yunbo Cao, and Zhoujun Li. 2021. Asking effective and diverse questions: A machine reading comprehension based framework for joint entity-relation extraction. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 3948–3954.

Xiaoyan Zhao, Min Yang, Qiang Qu, and Ruifeng Xu. 2024. Few-shot relation extraction with automatically generated prompts. *IEEE Transactions on Neural Networks and Learning Systems*, 36(3):4971–4983.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2022. Large language models are human-level prompt engineers. In *The eleventh international conference on learning representations*.

A Detailed Experimental Settings

A.1 Case Study

To illustrate the evolution of the REPO (Reinforcement Learning-based Automated Prompt Optimization) framework, we present a case study using the LCN dataset, tracing the complete lifecycle of a prompt from initial construction to deep reinforcement learning (DRL) refinement.

We employed meta-prompts to guide the large language model in generating seed prompts. Multiple seed prompts constitute the set of candidate prompts for the relationship extraction task. We then use REPQS to evaluate each prompt in the candidate set, selecting the top k% highest-scoring prompts to obtain the optimal seed prompt set,p_{seed}

1. Meta-prompts

Based on the provided example input and the corresponding example output processed by the Large Language Model, understand the underlying task. Then, generate a relationship extraction prompt that will enable the model to complete this task in future instances.
Requirements:
1.Return only the prompt itself.
2.The prompt must cover three specific relationship categories: Production, Supply, and Composition.
[Appended with a randomly selected relationship extraction example from the LCN dataset, organized in the 'Example Input - Example Output' format.]

2. seed prompts example

When extracting relationships, focus on the following three categories: Production, Supply, and Composition.
Production: Identify the products or services produced by a company.
Supply: Identify the products or services provided by a company to other companies or industries.
Composition: Identify the constituent parts or key technologies of a company's products or services.

Next, the process enters the iterative optimization phase, progressively exploring superior formulations through conversational rewriting by the large language model. Once new prompts are generated, a fusion update mechanism is employed to merge the prompts from the current round with high-performing prompts from previous iterations, forming a new candidate prompt set. In each round, only the top k% of prompts are selected to proceed to the next iteration.

3. Top k% Prompts - Next Iteration

Prompt: Production: Identifies the relationship when a company produces a certain product or service. For example: "Company A produces Product B".
Supply: Specifies that a company supplies products or services to a specific customer, market, or field. For example: "Company A supplies Product B to Customer C".
Composition: Refers to the components or raw materials that constitute a specific product or service. For example: "Product A is composed of Material B and Component C".
Examples: (Company Name, "Production", Product/Service)
(Company Name, "Supplier", Customer/-Market)
(Product/Service, "Composition", Raw Material/Component)

4. Non-Top $k\%$ Prompts - Excluded

Extract relationships involving Production, Supplier, and Composition from the text. Identify companies and the manufacturers, suppliers, or partners associated with their products, services, or equipment, and label the relationship types (e.g., "Production", "Supplier", "Composition", etc.).
For instance: (Company A, "Production", Product B), (Company A, "Supplier", Company C).

Upon completion of the iterations, the globally optimal prompt is returned as the output of the initial prompt construction phase.

5. initial prompt - the globally optimal prompt

You are a relationship extraction model. Based on the provided company-related text, please extract "Company-Relationship-Entity" triplets.
The requirements are as follows:
Extract only factual relationships explicitly stated in the text; do not make inferences or extensions.
Relationship types are restricted to the following three categories:
Produce: Products, technologies, or solutions that the company produces, researches, provides, specializes in, or owns.
Supply: The company acts as a provider or supplier, offering products or services to downstream customers, institutions, or enterprises.
Compose: A product, solution, or system is made up of multiple components, or the elements that constitute the company's business or solutions.
The output format must be: (Subject, "Relationship Type", Object). Separate multiple results with Chinese commas.
The subject is generally the company name, and the object consists of noun phrases such as specific products, customers, or components.
Do not output any explanations or descriptions; output only the extraction results.

After obtaining the initial prompt, enter the prompt optimization phase. Example: Select action 8 from the action space: Add input/output pairs.

6. Prompt After Action 8

You are a relationship extraction model. Based on the provided company-related text, please extract "Company-Relationship-Entity" triplets.
The requirements are as follows:
Extract only factual relationships explicitly stated in the text; do not make inferences or extensions.

6. Prompt After Action 8

Relationship types are restricted to the following three categories:
Produce: Products, technologies, or solutions that the company produces, researches, provides, specializes in, or owns.
Supply: The company acts as a provider or supplier, offering products or services to downstream customers, institutions, or enterprises.
Compose: A product, solution, or system is made up of multiple components, or the elements that constitute the company's business or solutions.
The output format must be: (Subject, "Relationship Type", Object). Separate multiple results with Chinese commas.
The subject is generally the company name, and the object consists of noun phrases such as specific products, customers, or components.
Do not output any explanations or descriptions; output only the extraction results.
Example Input: Caijianshe April 1st, an investor asked: Xiaomi SU7 has been launched; does the company currently have any cooperation with Xiaomi Auto? The company previously stated it cooperates with car manufacturerscould you disclose which ones? Qushui Technology stated on the interactive platform: As an internet retail company focusing on self-branded tech-innovative home products, the company is actively developing in-car home products. The company's newly developed products, such as in-car sunshades, have been launched for sale one after another, and the company has launched cooperation with some domestic car manufacturers; Xiaomi Auto is one of the company's important customers.
Output: (Qushui Technology, "Supply", Xiaomi), (Xiaomi, "Produce", Xiaomi Auto), (Qushui Technology, "Produce", In-car sunshades), (In-car sunshades, "Compose", Xiaomi Auto)

The best prompt is obtained after the prompt optimization phase is completed.

7. best prompt

Based on the content of the following news, summarize the Supply, Produce, and Compose relationships found within the text. Output the results in the form of multiple triplets. If no such relationship exists, output "None". The relationships are described as follows:
Supply Relationship: If the sentence describes Company A providing products to Company B, extract (A, "Supply", B).
Produce Relationship: If the sentence describes Company A producing a certain product P, extract (A, "Produce", P).
Compose Relationship: If the sentence describes product P as a component of product X, extract (P, "Compose", X).
The Supply relationship typically satisfies the following sentence structures:
1.Company A (provides / supplies) product P (to / for) Company B: extract (A, "Supply", B), (A, "Produce", P).
2.Company A provides product P, which is applied to Company B's products: extract (A, "Supply", B), (A, "Produce", P).
3.Company A is a supplier of Company B and provides product P to Company B: extract (A, "Supply", B), (A, "Produce", P).
4.Company B is a customer of Company A: extract (A, "Supply", B).
5.Company A supplies Company B: extract (A, "Supply", B).
Example Input:Cailianshe April 1st, an investor asked: Xiaomi SU7 has been launched; does the company currently have any cooperation with Xiaomi Auto? The company previously stated it cooperates with car manufacturerscould you disclose which ones? Qushui Technology stated on the interactive platform: As an internet retail company focusing on self-branded tech-innovative home products, the company is actively developing in-car home products. The company's newly developed products, such as in-car sunshades, have been launched for sale one after another, and the company has launched cooperation with some domestic car manufacturers; Xiaomi Auto is one of the company's important customers.
Output: (Qushui Technology, "Supply", Xiaomi), (, "Produce", Xiaomi Auto), (Qushui Technology, "Produce", In-car sunshades), (In-car sunshades, "Compose", Xiaomi Auto)

A.2 Cost Analysis

A comprehensive comparison of costs is displayed in Table 5. We demonstrate that while REPO requires an initial training overhead, the resulting converged optimal prompt (p^*) is highly robust and performs consistently in inference, reducing the need for the multiple trial-and-error attempts often required by static methods in complex domain-specific tasks. Also, we would like to point that the financial cost is no more than $50.

Table 5: Experimental Cost and Time of Different Methods on Multiple Datasets

Dataset	Method	Model	Token	API Call	Cost($)	Time(h)
CMeIE	APE	GPT-4o	106,113	84	0.69	2
	OPRO	GPT-4o	701,973	200	8.60	2
	SPO	GPT-4o	32028	120	0.27	1
	REPO	GPT-3.5-turbo(step 1)	2,944,477	3,957	27.32	24
		GPT-4o(step2)	1,018,022	476		
LexEval	APE	GPT-4o	69,137	72	0.50	1
	OPRO	GPT-4o	423433	123	5.18	2
	SPO	GPT-4o	40782	120	2.02	1
	REPO	GPT-3.5-turbo(step 1)	2,898,111	3,178	16.48	20
		GPT-4o(step2)	558,651	539		
FinCUGE	APE	GPT-4o	94,973	77	0.63	1
	OPRO	GPT-4o	643,339	173	7.88	3
	SPO	GPT-4o	20,571	80	1.02	1
	REPO	GPT-3.5-turbo(step 1)	2,728,908	3,499	25.04	26
		GPT-4o(step2)	889,933	915		
LCN	APE	GPT-4o	187,393	138	1.10	2
	OPRO	GPT-4o	1,042,442	297	12.71	5
	SPO	GPT-4o	64464	160	1.51	1
	REPO	GPT-3.5-turbo(step 1)	3,372,719	4,458	40.57	30
		GPT-4o(step2)	1,573,073	1,008		