RUIE: Retrieval-based Unified Information Extraction using Large Language Model

Xincheng Liao, Junwen Duan*, **Yixi Huang, Jianxin Wang**

Hunan Provincial Key Lab on Bioinformatics, School of Computer Science and Engineering,
Central South University, Changsha, Hunan, China
{ostars, jwduan, yx.huang}@csu.edu.cn, jxwang@mail.csu.edu.cn
https://github.com/OStars/RUIE

Abstract

Unified information extraction (UIE) aims to extract diverse structured information from unstructured text. While large language models (LLMs) have shown promise for UIE, they require significant computational resources and often struggle to generalize to unseen tasks. We propose RUIE (**R**etrieval-based **U**nified **I**nformation **E**xtraction), a framework that leverages in-context learning for efficient task generalization. RUIE introduces a novel demonstration selection mechanism combining LLM preferences with a keyword-enhanced reward model, and employs a bi-encoder retriever trained through contrastive learning and knowledge distillation. As the first trainable retrieval framework for UIE, RUIE serves as a universal plugin for various LLMs. Experimental results on eight held-out datasets demonstrate RUIE's effectiveness, with average F1-score improvements of 19.22 and 3.22 compared to instruction-tuning methods and other retrievers, respectively.

1 Introduction

Unified Information Extraction (UIE) represents a paradigm shift from traditional task-specific approaches, aiming to extract diverse structured information (e.g., Named Entity Recognition, Relationship Extraction, and Event Extraction) using a single model or framework. This unified approach demonstrates superior generalization capabilities and practical utility compared to conventional methods that require separate models for different extraction tasks (Lu et al., 2022; Wang et al., 2023b; Li et al., 2024). With the emergence of large language models (LLM) and their remarkable generalization abilities in various tasks (Wei et al., 2022; Zhang et al., 2024; Wang et al., 2024b; Jia et al., 2024), researchers have begun to explore LLM-based solutions to UIE challenges through
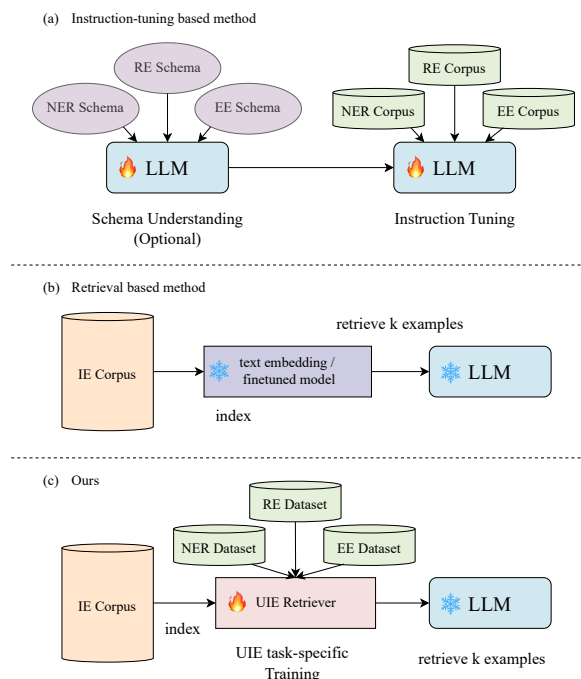

Figure 1: Illustration of three different paradigms for solving unified information extraction task.

two main approaches: instruction tuning and in-context learning (Figure 1).

However, the inherent mismatch between structured information extraction outputs and LLM's pretraining format poses significant challenges, resulting in current large models still underperforming specialized IE approaches (Han et al., 2023; Ma et al., 2023; Chen et al., 2024). Although some researchers (Wang et al., 2023b; Xiao et al., 2024b; Gui et al., 2024; Sainz et al., 2024; Li et al., 2024) have bridged this gap by transforming IE annotations into textual or code pairs for instruction tuning, this approach faces several critical limitations: substantial computational costs, potential degradation of general capabilities (Xu et al., 2024), and limited generalization to unseen tasks.

Alternatively, in-context learning (Brown et al., 2020) has emerged as a promising direction, allow-

---

ing LLMs to perform tasks with minimal examples or demonstrations. Recent works have demonstrated its effectiveness in IE tasks through various approaches: representing structured information in code format (Li et al., 2023; Wang et al., 2023c), improving extraction through offline sentence embedding and example retrieval (Guo et al., 2023), and developing task-specific semantic representations (Wang et al., 2023a; Wan et al., 2023). Despite these advances, existing retrieval-based methods remain largely task-specific, lacking true UIE capabilities, and primarily relying on semantic relevance while overlooking LLMs' inherent preferences in example selection.

To address these limitations while leveraging LLMs' capabilities, we propose RUIE, a novel retrieval-based unified information extraction framework. As illustrated in Figure 1, RUIE significantly reduces computational costs compared to instruction-tuning approaches by only requiring the fine-tuning of a smaller dense retriever (million-level parameters). Unlike existing retrieval-based methods, RUIE achieves true unified information extraction by maintaining a diverse candidate pool spanning multiple IE tasks (NER, RE, EE) and incorporating both semantic relevance and LLM preferences in example selection. Furthermore, we introduce a keyword-enhanced reward model to capture the label and fine-grained information, addressing the detailed nature of IE tasks (Wang et al., 2023a; Wan et al., 2023; Duan et al., 2024a,b). Our main contributions are as follows.

- We propose RUIE, a trainable retrieval framework for UIE that enables efficient task generalization through in-context learning while reducing computational costs.

- We develop an innovative demonstration selection mechanism that uniquely combines LLM preferences with a keyword-enhanced reward model, enabling more accurate and context-aware example selection.

- We demonstrate RUIE's strong flexibility as a general IE framework, showing robust performance across different tasks (NER, RE, EE) and easy integration with various LLM architectures.

2 Related Works

2.1 Unified Information Extraction

(Lu et al., 2022) first proposed the framework of unified Information Extraction. They used structured extraction language to unify the input and output forms of IE tasks and the structural schema instructor to guide the model generation, but it requires further fine-tuning for different downstream tasks. (Lou et al., 2023) considers the knowledge transfer between different tasks and schemas, modeling UIE as a semantic matching task with three dimensions of token-token, token-label and label-token, which achieves better zero/few-shot transfer ability. However, performing three-dimensional semantic matching for each token greatly increases the training and inference cost. Recently, some researchers have modeled IE as natural language (Wang et al., 2023b; Gui et al., 2024) or code (Sainz et al., 2024; Li et al., 2024) generation tasks. They constructed instruction-tuning datasets based on existing IE datasets to fine-tune large language models, realizing knowledge sharing of different tasks and effectively improving UIE performance. However, fine-tuning large language models is expensive, and the fine-tuned models do not generalize well in new domains.

2.2 In-context Learning based Information Extraction

In-context learning (ICL) is the ability of large language models to perform new tasks with only a few examples or demonstrations. One significant merit of ICL is the circumvention of fine-tuning, which might not always be possible due to limited access to the model parameters or constraints on computational resources (Brown et al., 2020). (Li et al., 2023; Wang et al., 2023c) represent structured IE tasks with codes, and improve extraction performance by randomly selecting demonstrations. (Guo et al., 2023) optimizes the random retrieval process by introducing sentence embedding to select demonstrations according to the similarity between query and demonstrations. (Wang et al., 2023a; Wan et al., 2023) show that fine-grained alignment information such as entities and relations is more important than sentence similarity for example selection in IE tasks. They use the entity and relation representations obtained by the fine-tuned small model to replace the sentence representation for retrieval, and obtain better performance than sentence-level embedding. However, they must obtain the entity span in the sentence and the fine-tuned small model in advance, which restricts the use cases. To the best of our knowledge, there is no trainable retrieval-based framework designed
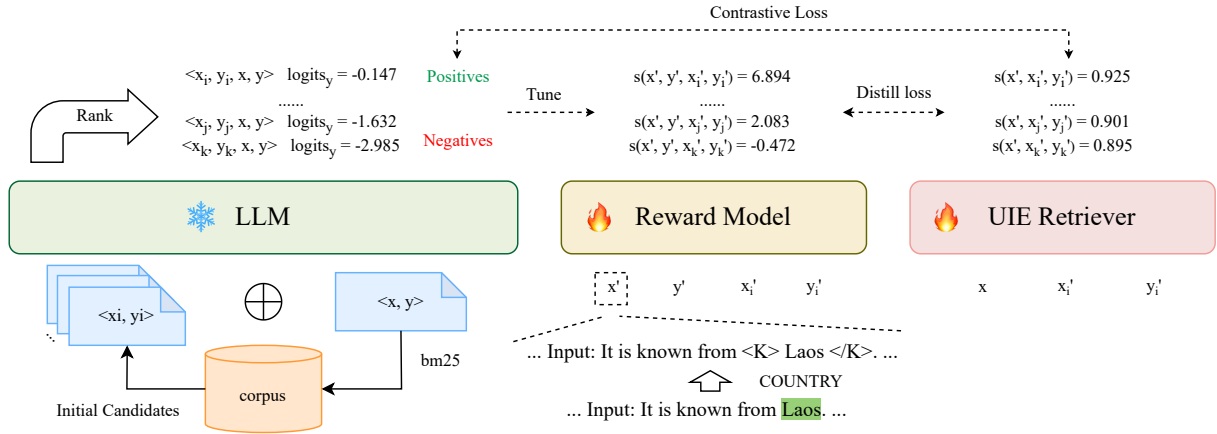
Figure 2: The overall architecture of RUIE. The training process consists of three steps: 1) the sparse retriever bm25 initializes a candidate set, which is then scored by the LLM. 2) a keyword-enhanced reward model captures fine-grained information. Keyword-enhanced strategy only applies to the input field of the example. 3) a bi-encoder dense retriever is trained using contrastive learning and knowledge distillation. During inference, the trained dense retriever selects the best demonstrations from the candidate pool P, and passes them to the LLM to produce the output.

for UIE. On the one hand, we do not need any prior information about the input sentence, on the other hand, we design a retrieval training scheme for UIE, which achieves better performance than general sentence embedding.

3 Methods

In this section, we first present the formal definition of information extraction and the task setting for retrieval-based UIE. Then we present our training framework illustrated in Figure 2.

3.1 Problem Statement

IE involves three main tasks: Named Entity Recognition (NER), Relation Extraction (RE), and Event Extraction (EE). For a given sentence x, NER seeks to extract tuples $\{s, e\}$, where s represents the entity span and e denotes the entity type. RE focuses on extracting triples $\{e_s, e_t, r\}$, with r being the relation type and e_s and e_t indicating the head and tail entity, respectively. EE comprises two sub-tasks: Event Detection (ED) and Event Argument Extraction (EAE). ED involves extracting event triggers $t \in \mathcal{O}$, where $\mathcal{O}$ is the event type ontology, while EAE extracts arguments $a \in \mathcal{R}$ for a given event trigger t, with $\mathcal{R}$ being the role type ontology.

Given a target task (such as NER) test sample x_{test} and k demonstrations $(x_i, y_i)_{i=1}^k$, we use a frozen large language model to generate answer y_{test} auto-regressively. Our goal is to retrieve k demonstrations from the candidate pool P that most closely match y_{test} to y_{truth}. It is worth noting that our candidate pool P contains a variety of information extraction tasks, such as NER, RE, ED and EAE.

3.2 LLM Preference Scoring

Sample Format. In the candidate pool P, each task sample (x_i, y_i) consists of four parts: (1) Task Name: name of a specific IE task, such as "Named Entity Recognition". (2) Schema: task ontology presented in the form of python list. (3) Input: input context to be extracted. (4) Output: structured output linearized by natural language, such as "Entitytype1: EntityName1; ...". (see Appendix D)

Score examples using LLM. Previous retrieval-based information extraction methods usually only consider the text similarity between the input text and the candidate pool examples, ignoring the preference of the LLM. We assume that the LLM knows what good examples are; specifically, good examples can maximize the probability that the model produces ground-truth y_{truth}. Therefore, given an input $s = (x, y)$ in the training set, we enumerate each example $e_i = (x_i, y_i)$ in the candidate pool P. To keep with the inference stage, we concatenate the extraction instruction I (see Appendix E), example e_i, and input s into the model. We serve the token-level average log-likelihood of y as the score between input s and example e_i:

$$Score(s, e_i) = \log p(y|I; x_i; y_i; x) \quad (1)$$

Then we rank the all candidate examples in descending order and select top k and last n as positive and negative examples respectively.

Initialize candidates. In practice, the candidate pool P often comprises a significantly large number of examples. To reduce the costs of LLM scoring, given an input sample $s = (x, y)$, we use a sparse retriever bm25 (Robertson and Zaragoza, 2009) to retrieve top-k candidate samples $\{(x_i, y_i)\}_i^k$ in the candidate pool P as the set of scoring candidates. Choosing an appropriate k can greatly improve the scoring efficiency and ensure that the candidate set contains both positive and negative samples.

3.3 Keyword-enhanced Reward

(Wang et al., 2023a; Wan et al., 2023) have shown that the alignment of query and candidate examples on fine-grained information such as entities and relations is more important than the alignment on coarse-grained information, like sentence semantics. Therefore, in order to make the fine-grained information between the input query and the candidate examples fully interactive, we propose a keyword-enhanced training strategy based on cross-encoder. On the one hand, the strategy aligns the fine-grained information through keyword, and on the other hand, the cross-encoder realizes the full interaction between query and candidate examples by accessing the ground-truth. Specifically, given a training sample (x, y), for each information snippet (sp, o) in y, where o is the label of span sp, we add a pair of special tags "<Keyword>" and "</Keyword>" around sp in context x (Figure 2). Given an enhanced input (x', y'), we sample a positive example (x'_+, y'_+) in the top-k of the enhanced ranked candidates, take the last-n as the negative examples (x'_{-i}, y'_{-i}) and produce a real-valued score s. We train the cross-encoder using the cross-entropy loss:

$$\mathcal{L}_{reward} = -log \frac{e^{s(x',y',x'_+,y'_+)}}{e^{s(x',y',x'_+,y'_+)} + \sum_{i=1}^{n} e^{s(x',y',x'_{-i},y'_{-i})}}$$
(2)

3.4 UIE Retriever Training

In the test phase, we have no access to the ground-truth output y corresponding to the input x. So we cannot directly use the above cross-encoder as the retriever. To enhance retrieval efficiency, we construct UIE retriever based on bi-encoder architecture. Specifically, we encode all samples

$e'_i = (x'_i, y'_i)$ in the keyword-enhanced candidate pool P into vector $h_{e'i}$ and build the index in advance. Given an input x, we use the same encoder to encode it into a vector h_x and calculate matching score between the input x and each example e'_i using temperature-scaled dot product:

$$score(x, e') = h_x \cdot h_{e'i}/\tau$$
$$h_{e'i} = \text{avgpool}(Encoder(x'_i, y'_i)) \quad (3)$$
$$h_x = \text{avgpool}(Encoder(x))$$

where "avgpool" refers to average pooling and τ refers to temperature hyperparameter.

We use two kinds of supervision signals to train the UIE retriever: (1) In order to make full use of the positive and negative pairs discovered by LLM, we use the Info-NCE loss $L_{contrastive}$ to perform contrastive learning between the top-k positives and in-batch negatives. (2) In order to make full use of the fine-grained alignment information of the reward model, we use the KL divergence to align the output distributions of the reward model and the retriever, calculated as $L_{distill}$, and the final training loss is the weighted sum of the above two losses:

$$\mathcal{L}_{retriever} = \mathcal{L}_{distill} + \alpha \mathcal{L}_{contrastive} \quad (4)$$

where α is a hyperparameter to measure the importance of the above two losses.

4 Experiment Setup

4.1 Datasets

To exhaustively evaluate the generalization ability of RUIE, we collect 31 held-in and 8 held-out datasets. We used the training set of the held-in and held-out datasets to form the candidate pool P. We constructed the retriever training set based on the held-in dataset, specifically, we sampled 10000 samples from each dataset and included all examples from datasets with less than 10000 samples. The Held-out dataset samples was completely unseen during training, and the UIE Retriever was tested on the test set of held-in and held-out datasets after training.

4.2 Metrics

We employ span-based Micro-F1 to evaluate the performance of our method. For NER, an entity is considered correct if the entity span and type are correctly predicted. For RE, a relation is considered correct if relation type, subject entity, and object

Method	NER	RE			ED / EAE			
	CrossNER	FewRel	Wiki-ZSL	#Avg	WikiEvents	RAMS	CrudeOil News	#Avg
Supervised Fine-tuning Methods								
UIE	38.37	-	-	-	5.12 / 1.78	9.25 / 2.14	6.45 / 8.95	6.94 / 4.29
InstructUIE	49.36	39.55	35.20	37.38	11.64 / 5.88	24.27 / 6.21	23.26 / 21.78	19.72 / 11.29
YAYI-UIE	50.39	36.09	41.07	38.58	10.97 / 5.11	18.87 / 8.21	12.45 / 19.74	14.10 / 11.02
LLaMA2-IEPILE	56.50	37.14	36.18	36.66	13.93 / 12.55	23.62 / 11.30	33.87 / 18.47	23.81 / 14.11
Retrieval-based Methods (k-shot=8)								
Random*	56.61	21.58	23.27	22.42	61.94 / 35.92	19.12 / 22.94	22.16 / 29.73	34.41 / 29.53
BM25*	63.62	44.86	49.88	47.37	63.99 / 43.78	33.42 / 25.67	49.96 / 53.04	49.12 / 40.83
E5	45.21	41.50	44.62	43.06	66.09 / 41.18	26.78 / 25.49	46.47 / 53.12	46.45 / 39.93
BGE	49.69	47.30	51.07	49.19	65.40 / 43.12	29.38 / 25.35	48.05 / 53.33	47.61 / 40.60
BM25	59.57	44.27	50.72	47.50	63.79 / 40.11	25.46 / 27.17	54.05 / 53.48	47.77 / 40.25
RUIE	65.41	49.93	53.16	51.55	66.42 / 40.64	34.60 / 26.06	51.50 / 53.79	50.84 / 40.16
RUIE-Deepseek	**69.60**	**57.57**	**60.14**	**58.85**	**71.88 / 44.30**	**47.72 / 38.51**	**68.73 / 61.04**	**62.77 / 47.95**

Table 1: Performance (in F1-score) on NER, RE, ED and EAE tasks under held-out settings. Bold indicates the highest scores and the second-best scores are underlined. "BM25*" and "Random*" refer to using BM25 or randomly selecting demonstrations from the candidate pool corresponding to the task, rather than from the multi-task candidate pool. The only difference between RUIE and other retrieval-based methods is that they use different retrievers. It's worth noting that the model size of InstuctUIE and LLaMA2-IEPILE are 11B and 13B, respectively, while the default model size of RUIE is 8B.

entity match the golden annotation. For EE task, we report two evaluation metrics: (1) ED: an event trigger is correct if the event type and the trigger word are correctly predicted. (2) EAE: an event argument is correct if its role type and event type match a reference argument mention.

4.3 Baseline Methods

We compare our approach with three categories of methods:

Non-generative UIE methods: (Lu et al., 2022) proposed a unified framework based on medium-sized language models with task-specific instructions and structured prediction.

Instruction-tuning methods: These approaches fine-tune large language models with task-specific instructions. (Wang et al., 2023b) reformulated the IE tasks using natural language instructions. (Gui et al., 2024) and (Xiao et al., 2024b) developed large-scale instruction datasets with JSON-formatted outputs. (Li et al., 2024) encoded IE structures through code-like formats.

Retrieval-based methods: We include both sparse and dense retrievers. For sparse retrieval, we use BM25 (Robertson and Zaragoza, 2009). For dense retrieval, we employ state-of-the-art embedding models E5 (Wang et al., 2024a) and BGE (Xiao et al., 2024a), which are trained through contrastive learning on large-scale text pairs. We exclude methods that require pre-identified entity

spans (Wang et al., 2023a; Wan et al., 2023) to ensure fair comparison under our end-to-end extraction setting.

4.4 Implementation Details

We employ LLaMA3-8B as the scoring model and LLaMA3.1-8B-instruct for inference in our experiments by default. For each test instance, we retrieve 8 demonstrations by default. The keyword-enhanced reward model and retriever are implemented based on ELECTRA-base and E5-base, respectively. Detailed hyperparameters and training configurations are provided in Appendix B.

5 Results and Analyses

5.1 Generalization to Held-out Tasks

We evaluate the generalization capability of different approaches on held-out tasks. The SFT-based methods are directly tested on held-out tasks after training on held-in tasks. RUIE first trains on held-in tasks, then retrieves demonstrations from the mixed candidate pool for LLM on held-out tasks. The remaining retrieval-based methods directly retrieve demonstrations from the mixed candidate pool for LLM on held-out tasks. The results are presented in Table 1.

RUIE achieves the best performance in four information extraction tasks, and demonstrates rapid generalization to new tasks. Compared

with SFT-based methods, the generalization ability of RUIE is significantly better. The performance of SFT-based methods decreases seriously on datasets not seen during training, with the drop becoming more pronounced as task difficulty increases. For example, in the ET and EAE tasks, the best performing LLaMA2-IEPILE model only achieves F1 scores of 23.81 and 14.11, making it impractical for use in the real world. In contrast, RUIE delivers improvements of 8.91, 14.89, 27.03, and 26.05 on NER, RE, ET, and EAE tasks, respectively, despite using a smaller model.

RUIE retrieves higher-quality examples compared to general-purpose retrievers. BM25 is a strong baseline and outperforms semantic similarity-based retrieval methods on average, confirming that fine-grained information alignment is more important in IE example retrieval. By explicitly modeling fine-grained information, RUIE introduces LLMs preference as a supervision signal during retrieval training. Compared to BM25, which has the best overall performance, RUIE achieves 5.84, 4.05, and 3.07 improvements on the NER, RE, and EET tasks, respectively.

RUIE demonstrates superior performance compared to simpler single-task corpus setups. When using general retrievers to retrieve examples for NER, there is a risk of retrieving RE or EE examples. In contrast, RUIE effectively minimizes the likelihood of retrieving examples from unrelated tasks. We conducted an experiment using BM25 to search within the candidate pool of the same task. Despite this simpler setup, RUIE achieved improvements of 1.79, 4.18, and 1.72 on the NER, RE, and EET tasks, respectively, even in multi-task candidate pool.

However, RUIE does not achieve the performance improvement on EAE. We believe there are two main reasons. First, compared to other IE tasks, event extraction data is more scarce, leading to insufficient training during the pre-training and alignment stages. Additionally, EAE requires the model to extract event parameters while understanding the event structure, making it more challenging than other tasks. Furthermore, the amount of EAE training data is less than other subtasks, limiting RUIE's generalization performance on new EAE tasks.

5.2 Results on Held-in Tasks

In the held-in tasks experiment, both the supervised fine-tuning methods and RUIE are trained on held-in tasks and tested on their corresponding

Dataset	SOTA	E5	BM25	RUIE
ACE 2004	▼ 87.60	42.79	48.46	56.53
ACE 2005	★ 86.66	42.53	48.35	55.86
AnatEM	★ 90.89	41.71	46.85	51.58
bc2gm	★ 85.16	43.48	45.99	49.78
bc4chemd	♠ 90.56	48.29	50.39	55.03
bc5cdr	★ 89.59	71.41	72.76	74.49
Broad Tweet	▲ 83.52	61.92	56.83	69.25
CoNLL 2003	▲ 96.77	68.09	68.50	78.34
FabNER	♣ 82.90	35.24	39.61	38.51
FindVehicle	♠ 99.45	68.86	73.53	92.26
GENIA_NER	♠ 78.29	53.23	57.64	59.85
HarveyNER	★ 88.79	29.74	33.59	37.72
MultiNERD	♣ 96.10	82.50	81.71	88.97
ncbi	★ 90.23	54.60	59.14	58.81
Ontonotes	★ 90.19	48.31	49.85	62.94
PolyglotNER	▲ 70.85	49.15	49.21	53.43
TweetNER7	▲ 66.99	49.93	49.79	53.17
WikiANN en	♣ 87.00	64.07	58.27	68.81
WikiNeural	★ 91.36	74.58	73.28	81.35
Avg	86.99	54.23	55.99	62.46

Table 2: Performance (in F1-score) comparison on NER tasks under held-in setting. ▼ indicates (Lu et al., 2022), ★ indicates (Wang et al., 2023b), ♠ indicates (Gui et al., 2024), ▲ inicates (Xiao et al., 2024b), ♣ indicates (Li et al., 2024).

test sets. The remaining retrieval-based methods directly retrieve k examples from held-in tasks for LLM. SFT-based methods include (Lu et al., 2022; Wang et al., 2023b; Gui et al., 2024; Xiao et al., 2024b; Li et al., 2024). As some baselines only report results for specific IE tasks, we report the SOTA results of the above methods in each dataset, denoted as "SOTA" in the tables.

The results of NER, RE, ED and EAE are shown in Tables 2, 3, 4, and 5. We conclude two main findings: (1) SFT-basd methods provide LLMs with a comprehensive understanding of IE through training in large-scale IE datasets, achieving superior results on held-in tasks compared to RUIE, which utilizes a frozen LLM. However, RUIE achieved an F1 score of 92.26 on FindVehicle, showing the strong potential of retrieval-based UIE. (2) RUIE showed markedly superior performance to generic retrievers, with improvements of 6.47, 7.13, 1.35, and 4.98 on the NER, RE, ED, and EAE tasks, respectively. This shows the effectiveness of introducing large model preference and keyword enhancement during the retriever training process.

Dataset	SOTA	E5	BM25	RUIE
ADE corpus	♠ 82.31	70.39	65.44	71.24
Conll04	★ 78.48	41.42	44.52	54.61
GIDS	★ 81.98	27.49	30.44	40.03
Kbp37	♣ 78.00	12.76	11.79	19.78
NYT	♠ 94.04	55.95	74.85	72.30
NYT11	▲ 57.53	31.16	33.49	40.92
SciERC	♠ 45.89	12.25	16.62	20.36
Semeval RE	★ 73.23	23.07	21.84	36.77
Avg	73.93	34.31	37.37	44.50

Table 3: Performance (in F1-score) comparison on RE tasks under held-in setting ★ indicates (Wang et al., 2023b), ♠ indicates (Gui et al., 2024), ▲ incicates (Xiao et al., 2024b), ♣ indicates (Li et al., 2024).

Dataset	SOTA	E5	BM25	RUIE
ACE2005	★ 77.13	42.90	39.06	53.41
CASIE	★ 67.80	29.35	39.69	40.46
PHEE	★ 70.14	37.18	59.21	47.16
Avg	71.69	36.48	45.99	47.01

Table 4: Performance (in F1-score) comparison on ED tasks under held-in setting. ★ indicates (Wang et al., 2023b).

We have also identified several key error patterns: 1. Domain knowledge gaps: Poor performance on medical datasets (e.g. bg2gm, SciERC) due to LLaMA3.1's limited medical domain training. 2. Label confusion: Difficulty distinguishing similar target labels (e.g. FabNER, kbp37). 3. Text formatting: Decreased precision with massive special symbols such as "@" and "#" in HarveyNER. 4. Pronoun handling: Missed pronoun entities in ACE2005. Moreover, we observed that the information extracted by the LLM and the actual labels only differ by one or two words, yet this does not hinder human understanding of the extraction results. This suggests that LLM-based information extraction methods require a more nuanced evaluation metric.

5.3 Ablation Study

We conducted an ablation study across all held-out datasets to underscore the effectiveness of the key innovations in our work (Table 6). Removing the keyword enhancement led to a 0.62 decrease in the average F1-score across four tasks, highlighting the value of fine-grained information alignment dur-

Dataset	SOTA	E5	BM25	RUIE
ACE2005	★ 72.94	39.06	38.29	44.29
CASIE	▲ 64.23	39.69	40.85	43.76
PHEE	▲ 77.19	59.21	59.10	65.13
Avg	71.45	45.99	46.08	51.06

Table 5: Performance (in F1-score) comparison on EAE tasks under held-in setting. ★ indicates (Wang et al., 2023b), ▲ incicates (Xiao et al., 2024b).

LLMs	NER	RE	ED / EAE
RUIE	68.24	**51.55**	**50.84 / 40.16**
- keyword-enhanced	**68.64**	51.37	48.82 / 39.47
- reward model	63.69	32.24	43.57 / 35.83
- distill loss	62.78	33.73	39.77 / 31.58

Table 6: Ablation study across all held-out datasets. "- keyword-enhanced" trains the reward model and retriever without keyword enhancement. "- reward model" trains the retriever with distill loss on LLMs scores and contrastive loss on ranked candidates. "- distill loss" trains the retriever soly with contrastive loss on ranked candidates.

ing retriever training. After removing distill loss, the average F1-score of the four tasks decreases by 10.73, which is because the rank of candidate examples only reflects the relative distribution of LLMs preferences. However, knowledge distillation based on KL-divergence can align the absolute distribution of LLMs preferences, which confirms the importance of using LLMs preferences as supervision signals. After removing the reward model, the average F1-score of the four tasks decreases by 8.87, indicating the log-likelihood from LLMs is not suitable as a direct knowledge source for distillation. The reason is that the average log-likelihood is not a true probability distribution, and its values tend to cluster within a narrow range, making it less effective as a target distribution within the KL-divergence framework.

5.4 The Effects of Different K-shot Demonstrations

We first used the default experimental setup to investigate the effects of varying k-shot demonstrations on extraction performance. Experiments were conducted across all held-in and held-out datasets, and we reported the average F1 scores for each task across all datasets, as depicted in Figure 3. We summarized three findings: (1) The performance across
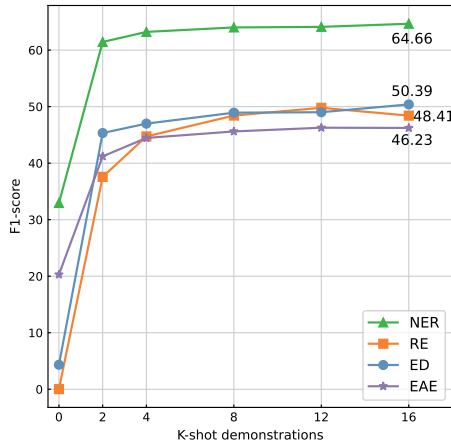
Figure 3: Performance (in F1-score) comparison by varying k-shot demonstrations.
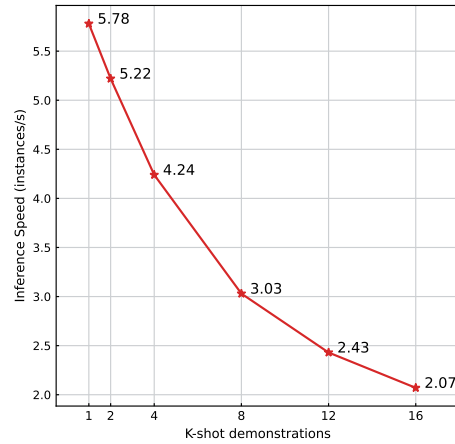

Figure 4: Inference speed (instances per second) comparison of different k-shots demonstrations on ACE 2004 (NER), ADE corpus (RE) and ACE2005 (ED and EAE) datasets.

all tasks improved with an increase in k, with the most notable enhancement occurring as k rose from 0 to 2, indicating that few-shot demonstrations are crucial for LLMs to tackle tasks effectively. (2) The task performance does not consistently improve with increasing k. For example, there is a performance drop when k increased from 12 to 16 in the RE task. We surmise that one contributing factor is the limitation of the model's context length, while another is the additional noise introduced by an excess of examples, which can adversely affect model performance. Therefore, we set k to 8, which can take into account both inference efficiency and extraction performance. (3) Model performance is influenced by the complexity of the task. NER has more data and is easier compared to other information extraction tasks, exhibited significantly better zero-shot performance than other tasks, while LLM was virtually incapable of completing the RE and ED tasks in a zero-shot scenario. Even as the number of examples k increased, the model's performance on NER remained superior to that of other information extraction tasks.

Then we used the default experimental setup to investigate the effects of different k-shots on inference efficiency. Since SFT-based methods with similar size do not support vLLM, we treat the case of k-shot = 1 as equivalent to the SFT-based methods for fairness (RUIE is built based on bi-encoder and the retrieval time of a single sample can be negligible in our experiment). As shown in Figure 4, the inference speed decreases consistently with increasing number of k shots, with k-shot = 1 approximately 1.9 times faster than k-shot = 8. While RUIE has lower inference efficiency than

SFT-based due to retrieval overhead and longer context length, it offers three key advantages: 1. Minimal training cost: Only fine-tunes a small retriever vs. full LLM. 2. Better domain adaptation: Using ICL with a small amount of labeled data vs. full retraining. 3. API compatibility: Works directly with LLM APIs vs. requiring model deployment.

5.5 The Effects of Different Scoring LLMs

LLMs	NER	RE	ED / EAE
GPTNeo	**68.27**	50.78	47.86 / 40.35
LLaMA3-instruct	68.14	51.24	49.07 / **40.68**
LLaMA3	68.24	**51.55**	**50.84** / 40.16

Table 7: Performance (in F1-score) comparison of different scoring LLMs on NER, RE, ED, EAE tasks across all held-out datasets.

To investigate the impact of various scoring LLMs on the extraction performance, we conducted experiments across all held-out datasets, with results presented in Table 7. We have two findings: 1) The size of the scoring LLM has a minor influence on the final performance. Although LLaMA3 (8B) outperformed GPTNeo (2.7B) by 2.98 on ED task, their performance was comparable on NER, RE and EAE tasks. This suggests that models of different sizes exhibit similar preferences for certain tasks, allowing one to select an appropriately sized LLM based on computational resources. 2) The base version of the LLM is more suitable as

a scoring model than the instruct version. We calculated the mean and variance of the scores for the positive samples (top-3) and the negative samples (last-16) for both versions of the LLM (Table 10). Although the mean difference between positive and negative sample scores of instruct version LLM is larger, the variance of positive and negative sample scores is significantly larger than that of base version LLM, which introduces instability to the subsequent training of the reward model, resulting in the final extraction performance inferior to base version.

5.6 Performance Analysis Across Different LLMs for Inference

Table 8 shows the performance of various reasoning LLMs across all held-out datasets. The ability of the inference LLMs significantly influences the extraction performance, which is expressed in two aspects: 1) Model size: For the same Qwen1.5, an increase in model size from 7B to 14B resulted in an average performance enhancement of 9.21 across four tasks. 2) Model type: Within the LLaMA series, LLaMA3.1, which is an advancement over LLaMA3, achieved an average performance improvement of 0.87 across four tasks. Since all tasks are in English, the LLaMA series models consistently outperformed the Qwen series. Additionally, RUIE adapts effectively to both locally deployed models and API-based models, with Deepseek demonstrating the best performance overall.

LLMs	NER	RE	ED / EAE
Qwen1.5-7b	54.70	33.86	33.84 / 26.90
Qwen1.5-14b	63.35	44.53	40.09 / 32.23
LLaMA3-8b	67.19	53.65	46.83 / 39.64
LLaMA3.1-8b	68.24	51.55	50.84 / 40.16
Deepseek	**72.24**	**58.85**	**62.77 / 47.95**

Table 8: Performance (in F1-score) comparison of different inference LLMs on NER, RE, ED, EAE tasks across all held-out datasets. Deepseek refers to the deepseek-chat-v2. All other models are instruct variant, which are further fine-tuned on instruction-tuning datasets after pre-training.

6 Conclusion

In this paper, we introduced RUIE, a novel trainable retrieval-based framework for unified information extraction that addresses key challenges in the field. Our framework introduced a pioneering trainable retrieval mechanism specifically designed for UIE tasks, significantly reducing computational costs while enabling rapid generalization to unseen tasks. Extensive experiments on 31 training datasets and 8 held-out tasks demonstrated RUIE's superior performance across various IE tasks, suggesting promising directions for developing more efficient and adaptable IE systems.

Limitations

Sequence length constraint: RUIE currently focuses on sentence-level UIE. On the one hand, it is difficult to meet the retrieval requirements of long documents due to the length limitation of the retriever is 512. On the other hand, due to the length limitation of LLMs, k-shot cannot be increased in the case of long documents, which limits the performance of the model.

Gap between RUIE and the SFT-based methods in seen tasks: The SFT-based methods inject UIE-specific knowledge into the LLMs through fine-tuning, while RUIE frozens the LLMs and can only leverage the knowledge from the pre-training phase. How to better stimulate the information extraction ability of LLMs under controllable cost is a worthy study direction.

English corpus only: RUIE is currently only trained and tested on English data. In the future, we hope to expand RUIE to more languages.

Acknowledgments

References

Rami Al-Rfou, Vivek Kulkarni, Bryan Perozzi, and Steven Skiena. 2014. Polyglot-ner: Massive multilingual named entity recognition. *Preprint*, arXiv:1410.3791.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu,

Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *CoRR*, abs/2005.14165.

Chih-Yao Chen and Cheng-Te Li. 2021. Zs-bert: Towards zero-shot relation extraction with attribute representation learning. *Preprint*, arXiv:2104.04697.

Pei Chen, Haotian Xu, Cheng Zhang, and Ruihong Huang. 2022. Crossroads, buildings and neighborhoods: A dataset for fine-grained location recognition. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3329–3339, Seattle, United States. Association for Computational Linguistics.

Ruirui Chen, Chengwei Qin, Weifeng Jiang, and Dongkyu Choi. 2024. Is a Large Language Model a Good Annotator for Event Extraction? *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(16):17772–17780. Number: 16.

Leon Derczynski, Kalina Bontcheva, and Ian Roberts. 2016. Broad Twitter corpus: A diverse named entity recognition resource. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1169–1179, Osaka, Japan. The COLING 2016 Organizing Committee.

Rezarta Islamaj Dogan, Robert Leaman, and Zhiyong Lu. 2014. Ncbi disease corpus: A resource for disease name recognition and concept normalization. *Journal of biomedical informatics*, 47:1–10.

Junwen Duan, Xincheng Liao, Ying An, and Jianxin Wang. 2024a. Keyee: Enhancing low-resource generative event extraction with auxiliary keyword subprompt. *Big Data Mining and Analytics*, 7(2):547–560.

Junwen Duan, Shuyue Liu, Xincheng Liao, Feng Gong, Hailin Yue, and Jianxin Wang. 2024b. Chinese emr named entity recognition using fused label relations based on machine reading comprehension framework. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 21(5):1143–1153.

Seth Ebner, Patrick Xia, Ryan Culkin, Kyle Rawlins, and Benjamin Van Durme. 2020. Multi-sentence argument linking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Runwei Guan, Ka Lok Man, Feifan Chen, Shanliang Yao, Rongsheng Hu, Xiaohui Zhu, Jeremy Smith, Eng Gee Lim, and Yutao Yue. 2023. Findvehicle and vehiclefinder: A ner dataset for natural language-based vehicle retrieval and a keyword-based cross-modal vehicle retrieval system. *Preprint*, arXiv:2304.10893.

Honghao Gui, Lin Yuan, Hongbin Ye, Ningyu Zhang, Mengshu Sun, Lei Liang, and Huajun Chen. 2024. IEPile: Unearthing large scale schema-conditioned information extraction corpus. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 127–146, Bangkok, Thailand. Association for Computational Linguistics.

Yucan Guo, Zixuan Li, Xiaolong Jin, Yantao Liu, Yutao Zeng, Wenxuan Liu, Xiang Li, Pan Yang, Long Bai, Jiafeng Guo, and Xueqi Cheng. 2023. Retrieval-Augmented Code Generation for Universal Information Extraction.

Harsha Gurulingappa, Abdul Mateen Rajput, Angus Roberts, Juliane Fluck, Martin Hofmann-Apitius, and Luca Toldo. 2012. Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports. *Journal of Biomedical Informatics*, 45(5):885–892. Text Mining and Natural Language Processing in Pharmacogenomics.

Ridong Han, Tao Peng, Chaohao Yang, Benyou Wang, Lu Liu, and Xiang Wan. 2023. Is information extraction solved by chatgpt? an analysis of performance, evaluation criteria, robustness and errors. *Preprint*, arXiv:2305.14450.

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. *Preprint*, arXiv:1810.10147.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2010. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In **SEMEVAL*.

Eduard H. Hovy, Mitchell P. Marcus, Martha Palmer, Lance A. Ramshaw, and Ralph M. Weischedel. 2006. Ontonotes: The 90% solution. In *North American Chapter of the Association for Computational Linguistics*.

Sharmistha Jat, Siddhesh Khandelwal, and Partha Pratim Talukdar. 2018. Improving distantly supervised relation extraction using word and entity based attention. *ArXiv*, abs/1804.06987.

Mingyi Jia, Junwen Duan, Yan Song, and Jianxin Wang. 2024. medikal: Integrating knowledge graphs as assistants of llms for enhanced clinical diagnosis on emrs. *Preprint*, arXiv:2406.14326.

Jin-Dong Kim, Tomoko Ohta, Yuka Tateisi, and Junichi Tsujii. 2003a. Genia corpus - a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19 Suppl 1:i180–2.

Jin-Dong Kim, Tomoko Ohta, Yuka Tateisi, and Jun'ichi Tsujii. 2003b. Genia corpus—a semantically annotated corpus for bio-textmining. *Bioinformatics (Oxford, England)*, 19 Suppl 1:i180–2.

Veysel Kocaman and David Talby. 2020. Biomedical named entity recognition at scale. In *ICPR Workshops*.

Martin Krallinger, Obdulia Rabal, Florian Leitner, Miguel Vazquez, David Salgado, Zhiyong lu, Robert Leaman, Yanan Lu, Donghong Ji, Daniel Lowe, Roger Sayle, Riza Batista-Navarro, Rafal Rak, Torsten Huber, Tim Rocktäschel, Sérgio Matos, David Campos, Buzhou Tang, Wang Qi, and Alfonso Valencia. 2015. The chemdner corpus of chemicals and drugs and its annotation principles. *Journal of Cheminformatics*, 7:S2.

Aman Kumar and Binil Starly. 2021. "fabner": information extraction from manufacturing process science domain literature using named entity recognition. *Journal of Intelligent Manufacturing*, 33:2393 – 2407.

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*, SOSP '23, page 611–626, New York, NY, USA. Association for Computing Machinery.

Meisin Lee, Lay-Ki Soon, Eu Gene Siew, and Ly Fie Sugianto. 2022. CrudeOilNews: An annotated crude oil news corpus for event extraction. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 465–479, Marseille, France. European Language Resources Association.

Jiao Li, Yueping Sun, Robin J. Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J. Mattingly, Thomas C. Wiegers, and Zhiyong Lu. 2016. Biocreative v cdr task corpus: a resource for chemical disease relation extraction. *Database: The Journal of Biological Databases and Curation*, 2016.

Peng Li, Tianxiang Sun, Qiong Tang, Hang Yan, Yuanbin Wu, Xuanjing Huang, and Xipeng Qiu. 2023. CodeIE: Large Code Generation Models are Better Few-Shot Information Extractors. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15339–15353, Toronto, Canada. Association for Computational Linguistics.

Sha Li, Heng Ji, and Jiawei Han. 2021. Document-level event argument extraction by conditional generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 894–908, Online. Association for Computational Linguistics.

Zixuan Li, Yutao Zeng, Yuxin Zuo, Weicheng Ren, Wenxuan Liu, Miao Su, Yucan Guo, Yantao Liu, Lixiang Lixiang, Zhilei Hu, Long Bai, Wei Li, Yidan Liu, Pan Yang, Xiaolong Jin, Jiafeng Guo, and Xueqi Cheng. 2024. KnowCoder: Coding structured knowledge into LLMs for universal information extraction. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8758–8779, Bangkok, Thailand. Association for Computational Linguistics.

Yijin Liu, Fandong Meng, Jinchao Zhang, Jinan Xu, Yufeng Chen, and Jie Zhou. 2019. GCDT: A global context enhanced deep transition architecture for sequence labeling. *CoRR*, abs/1906.02437.

Zihan Liu, Yan Xu, Tiezheng Yu, Wenliang Dai, Ziwei Ji, Samuel Cahyawijaya, Andrea Madotto, and Pascale Fung. 2021. Crossner: Evaluating cross-domain named entity recognition. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(15):13452–13460.

Jie Lou, Yaojie Lu, Dai Dai, Wei Jia, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. 2023. Universal information extraction as unified semantic matching. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'23/IAAI'23/EAAI'23. AAAI Press.

Yaojie Lu, Hongyu Lin, Jin Xu, Xianpei Han, Jialong Tang, Annan Li, Le Sun, M. Liao, and Shaoyi Chen. 2021. Text2event: Controllable sequence-to-structure generation for end-to-end event extraction. *ArXiv*, abs/2106.09232.

Yaojie Lu, Qing Liu, Dai Dai, Xinyan Xiao, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. 2022. Unified Structure Generation for Universal Information Extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5755–5772, Dublin, Ireland. Association for Computational Linguistics.

Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. *Preprint*, arXiv:1808.09602.

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

Alexis Mitchell, Stephanie Strassel, Shudong Huang, and Ramez Zakhary. 2005. Ace 2004 multilingual training corpus. *Linguistic Data Consortium, Philadelphia*, 1:1–1.

openbiocorpora. 2015. openbiocorpora anatem.

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. 2017. Cross-lingual name tagging and linking for 282 languages. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1946–1958, Vancouver, Canada. Association for Computational Linguistics.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *ECML/PKDD*.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Dan Roth and Wen tau Yih. 2004. A linear programming formulation for global inference in natural language tasks. In *Conference on Computational Natural Language Learning*.

Oscar Sainz, Iker García-Ferrero, Rodrigo Agerri, Oier Lopez de Lacalle, German Rigau, and Eneko Agirre. 2024. GoLLIE: Annotation guidelines improve zero-shot information-extraction. In *The Twelfth International Conference on Learning Representations*.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *Preprint*, arXiv:cs/0306050.

Zhao-Li Sun, Jiazheng Li, Gabriele Pergola, Byron C. Wallace, Bino John, Nigel Greene, Joseph Kim, and Yulan He. 2022. Phee: A dataset for pharmacovigilance event extraction from text. *ArXiv*, abs/2210.12560.

Ryuichi Takanobu, Tianyang Zhang, Jiexi Liu, and Minlie Huang. 2018. A hierarchical framework for relation extraction with reinforcement learning. In *AAAI Conference on Artificial Intelligence*.

Simone Tedeschi, Valentino Maiorca, Niccolò Campolungo, Francesco Cecconi, and Roberto Navigli. 2021. WikiNEuRal: Combined neural and knowledge-based silver data creation for multilingual NER. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2521–2533, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Simone Tedeschi and Roberto Navigli. 2022. MultiNERD: A multilingual, multi-genre and fine-grained dataset for named entity recognition (and disambiguation). In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 801–812, Seattle, United States. Association for Computational Linguistics.

Asahi Ushio, Leonardo Neves, Vitor Silva, Francesco Barbieri, and Jose Camacho-Collados. 2022. Named entity recognition in twitter: A dataset and analysis on short-term temporal shifts. *Preprint*, arXiv:2210.03797.

C. Walker and Linguistic Data Consortium. 2005. *ACE 2005 Multilingual Training Corpus*. LDC corpora. Linguistic Data Consortium.

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, Singapore. Association for Computational Linguistics.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2024a. Text embeddings by weakly-supervised contrastive pre-training. *Preprint*, arXiv:2212.03533.

Liang Wang, Nan Yang, and Furu Wei. 2024b. Learning to retrieve in-context examples for large language models. *Preprint*, arXiv:2307.07164.

Shuhe Wang, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, and Guoyin Wang. 2023a. GPT-NER: Named Entity Recognition via Large Language Models. *arXiv preprint*. ArXiv:2304.10428 [cs].

Xiao Wang, Weikang Zhou, Can Zu, Han Xia, Tianze Chen, Yuansen Zhang, Rui Zheng, Junjie Ye, Qi Zhang, Tao Gui, Jihua Kang, Jingsheng Yang, Siyuan Li, and Chunsai Du. 2023b. InstructUIE: Multi-task Instruction Tuning for Unified Information Extraction. *arXiv preprint*. ArXiv:2304.08085 [cs].

Xingyao Wang, Sha Li, and Heng Ji. 2023c. Code4Struct: Code Generation for Few-Shot Event Structure Prediction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3640–3663, Toronto, Canada. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *CoRR*, abs/2201.11903.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. 2024a. C-pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 641–649, New York, NY, USA. Association for Computing Machinery.

Xinglin Xiao, Yijie Wang, Nan Xu, Yuqi Wang, Hanxuan Yang, Minzheng Wang, Yin Luo, Lei Wang, Wenji Mao, and Daniel Zeng. 2024b. YAYI-UIE: A Chat-Enhanced Instruction Tuning Framework for Universal Information Extraction. *arXiv preprint*. ArXiv:2312.15548 [cs].

Jun Xu, Mengshu Sun, Zhiqiang Zhang, and Jun Zhou. 2024. ChatUIE: Exploring chat-based unified information extraction using large language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 3146–3152, Torino, Italia. ELRA and ICCL.

Chao Zhang, Shiwei Wu, Haoxin Zhang, Tong Xu, Yan Gao, Yao Hu, and Enhong Chen. 2024. Notellm: A retrievable large language model for note recommendation. In *Companion Proceedings of the ACM Web Conference 2024*, WWW '24, page 170–179, New York, NY, USA. Association for Computing Machinery.

Dongxu Zhang and Dong Wang. 2015. Relation classification via recurrent neural network. *Preprint*, arXiv:1508.01006.

A Data Details

Our candidate pool P mainly consists of IE INSTRUCTIONS (Wang et al., 2023b) and IEPILE (Gui et al., 2024). There are 22 NER datasets: ACE2004 (Mitchell et al., 2005), ACE2005 (Walker and Consortium, 2005), Broad Twitter (Derczynski et al., 2016), CoNLL2003 (Sang and Meulder, 2003), MultiNERD (Tedeschi and Navigli, 2022), Ontonotes (Hovy et al., 2006), Polyglot-NER (Al-Rfou et al., 2014), tweetNER7 (Ushio et al., 2022), wikiANN (Pan et al., 2017), wikineural (Tedeschi et al., 2021), AnatEM (openbiocorpora, 2015), bc2gm (Kocaman and Talby, 2020), bc4chemd (Krallinger et al., 2015), bc5cdr (Li et al., 2016), FabNER(Kumar and Starly, 2021), FindVehicle (Guan et al., 2023), GENIA (Kim et al., 2003b), HarveyNER (Chen et al., 2022), MIT Movie (Liu et al., 2019) MIT Restaurant (Liu et al., 2019) ncbi-disease (Dogan et al., 2014). For the RE task, we utilize 10 datasets including ADE corpus (Gurulingappa et al., 2012), CoNLL04 (Roth and tau Yih, 2004), GIDS (Jat et al., 2018), kbp37 (Zhang and Wang, 2015), NYT (Riedel et al., 2010), NYT11 HRL (Takanobu et al., 2018), SciERC (Luan et al., 2018), semeval RE (Hendrickx et al., 2010), FewRel (Han et al., 2018) and Wiki-ZSL (Chen and Li, 2021). For the EE task, ACE2005 (Walker and Consortium, 2005), CASIE(Lu et al., 2021), GENIA(Kim et al., 2003a), PHEE(Sun et al., 2022), CrudeOilNews (Lee et al., 2022), RAMS (Ebner et al., 2020) and WikiEvents (Li et al., 2021) are used.

B Implementation Details

We finished LLM scoring, reward and retriever training on two 3090 GPUs. In order to balance

	Reward	Retriever
learning rate	1e-5	3e-5
batch size	64	128
training steps	3000	6000
α	-	0.2
τ	-	0.01
positives	top-3	top-3
negatives	last-16	in-batch negatives
input length	512	512

Table 9: Hyperparameters for reward and retriever model training.

the efficiency and performance of the scoring process, we used bm25 to initialize 100 candidates for each sample in the training set. Due to the sequence length limitation of the LLMs, we retrieved 8 samples for each query and set the maximum input length to 1792 and the maximum generation length to 256. We performed inference using vLLM (Kwon et al., 2023) on a single 3090 GPU. In order to ensure the reproduction of the results, we use a greedy decoding strategy and set the temperature to 0.

C Score Mean and Variance of different LLMs

Model	Positives		Negatives	
	mean	variance	mean	variance
base	-0.39	0.18	-1.02	0.93
instruct	-0.36	0.35	-1.23	2.36

Table 10: Score Mean and Variance of different LLMs. "base" indicates LLaMA3-8B and "instruct" indicates LLaMA3-8B-instruct. We reported score mean and variance in positives and negatives separately.

D Sample Format

Detailed sample format is listed in Table 12.

E Instruction Format

Detailed instruction format is listed in Table 13.

Task	Dataset	#Schema	#Train	#Test	Training	Evaluation
NER	ACE2004	7	6202	812	√	√
	ACE2005	7	7299	1060	√	√
	Broad Tweet	3	5334	2001	√	√
	CoNLL2003	4	14041	3453	√	√
	multiNERD	16	134144	10000	√	√
	Ontonotes	18	59924	8262	√	√
	Polyglot-NER	3	393982	10000	√	√
	tweetNER7	7	7111	576	√	√
	Wikiann	3	20000	10000	√	√
	wikineural	3	92729	11597	√	√
	anatEM	1	5861	3830	√	√
	Bc2gm	1	12500	5000	√	√
	Bc4chemd	1	30682	26364	√	√
	Bc5cd	2	4560	4797	√	√
	FabNER	12	9435	2064	√	√
	FindVehicle	21	21565	20777	√	√
	GENIA	5	15023	1854	√	√
	HarveyNER	4	3967	1303	√	√
	Ncbi-disease	1	5432	940	√	√
	CrossNER AI†	14	-	431		√
	CrossNER Literature†	12	-	416		√
	CrossNER Music†	13	-	465		√
	CrossNER Politics†	9	-	650		√
	CrossNER Science†	17	-	543		√
	MIT Movie Review†	12	-	2442		√
	MIT Restaurant Review†	8	-	1520		√
RE	ADE corpus	1	3417	428	√	√
	CoNLL2004	5	922	288	√	√
	GIDS	4	8526	4307	√	√
	Kbp37	18	15917	3405	√	√
	NYT	24	56196	5000	√	√
	NYT11 HRL	12	62648	369	√	√
	SciERC	7	1366	397	√	√
	Semeval RE	10	6507	2717	√	√
	FewRel†	83	-	17291		√
	Wiki†	100	-	23113		√
EE	ACE2005	33(22)	3342	293	√	√
	CASIE	5(26)	3751	1500	√	√
	GENIA	5(0)	15023	1854	√	√
	PHEE	2(16)	2898	968	√	√
	CrudeOilNews†	18(104)	-	356		√
	RAMS†	106(398)	-	887		√
	WikiEvents†	31(81)	-	249		√

Table 11: Statistical data of all 39 IE datasets. Datasets with † are held-out datasets, which is unseen during retriever training stage. CrossNER (Liu et al., 2021) is divided into five subsets for our statistical analysis. For the held-out datasets, we only report their test set.

Task	Sample Format
NER	**Task:** Named Entity Recognition **Schema:** [location, person, organization] **Input:** The Parkinsons are a punk rock band originally from Coimbra, Portugal, formed in the year 2000 and based in London, known for their outrageous live performances. **Output:** location: Coimbra; location: Portugal; location: London.
RE	**Task:** Relation Extraction **Schema:** [Organization based in, Located in, Live in, Work for, Kill] **Input:** Washington: About 110 firefighters cut a containment line most of the way around an 850-acre forest fire in the Pasayten Wilderness near the Canadian border Tuesday. **Output:** Located in: Pasayten Wilderness, Washington.
ED	**Task:** Event Detection **Schema:** [phishing, data breach, ransom, discover vulnerability, patch vulnerability] **Input:** Google Project Zero's security researchers have discovered another critical remote code execution (RCE) vulnerability in Microsoft's Windows operating system, claiming that it is something truly bad. **Output:** discover vulnerability: have discovered.
EAE	**Task:** Event argument extraction **Schema:** Given event trigger: "discover vulnerability: discovered"; Candidate arguments: [vulnerable system owner, vulnerability, capabilities, time, vulnerable system version, discoverer, common vulnerabilities and exposures, supported platform, vulnerable system] **Input:** Google Project Zero's security researchers have discovered another critical remote code execution (RCE) vulnerability in Microsoft's Windows operating system, claiming that it is something truly bad. **Output:** vulnerable system: Windows operating system; vulnerability: remote code execution (RCE) vulnerability; discoverer: security researchers; vulnerable system owner: Microsoft; discoverer: Google Project Zero.

Table 12: Detailed sample format across four IE tasks. NER sample is from polyglotNER (Al-Rfou et al., 2014), RE sample is from conll04 (Roth and tau Yih, 2004), and both ED and EAE samples are from CASIE (Lu et al., 2021).

Task	Instruction
NER	Please analyze the given schema and extract all named entities from the provided input. Follow these instructions carefully: 1. Output Format: Present the extracted entities in this structured format: "EntityType1: EntityName1; EntityType2: EntityName2; ...". 2. Include Only Present Entities: Only output entities that actually exist in the input. Ignore any entities that are not mentioned. 3. No Entities Response: If the input contains no named entities, respond with "None". 4. Examples Handling: If examples are provided for reference, use them to understand the annotation criteria, but do not extract entities from these examples.
RE	Please analyze the given sentence and extract subjects and objects that have a specific relation, according to the provided schema. Follow these guidelines: 1. Output Format: Format your output as follows: "relation1: subject1, object1; relation2: subject2, object2; ...". 2. Include Only Present Relations: Only output relations that actually exist in the input. Ignore any relations that are not mentioned. 3. No Relations Response: If the sentence contains no relations, respond with "None". 4. Example Usage: If examples are provided, use them to understand the annotation criteria. Do not extract relations from these examples.
ED	Please analyze the given schema and extract all event triggers from the provided input. Follow these instructions carefully: 1. Output Format: Present the extracted triggers in this structured format: "EventType1: TriggerName1; EventType2: TriggerName2; ...". 2. Include Only Present Triggers: Only output triggers that actually exist in the input. Ignore any triggers that are not mentioned. 3. No Triggers Response: If the input contains no event triggers, respond with "None". 4. Examples Handling: If examples are provided for reference, use them to understand the annotation criteria, but do not extract triggers from these examples.
EAE	Please extract and list arguments of specified types for a given event type and trigger, according to the provided schema. Follow these instructions carefully: 1. Output Format: Format your output as follows: "ArgumentType1: argument1; ArgumentType2: argument2; ...". 2. Include Only Present Arguments: Only include event arguments that are present in the input sentence. Ignore any event arguments that are not mentioned. 3. No Arguments Response: If the sentence contains no event arguments, respond with "None". 4. Example Usage: If examples are provided, use them to understand the annotation criteria. Do not extract event arguments from these examples.

Table 13: Detailed instruction format across four IE tasks.