Recall, Retrieve and Reason: Towards Better In-Context Relation Extraction

Guozheng Li[1], Peng Wang[1,2*], Wenjun Ke[1,2], Yikai Guo[3],
Ke Ji[1], Ziyu Shang[1], Jiajun Liu[1] and Zijie Xu[1]
[1]School of Computer Science and Engineering, Southeast University
[2]Key Laboratory of New Generation Artificial Intelligence Technology and Its
Interdisciplinary Applications (Southeast University), Ministry of Education
[3]Beijing Institute of Computer Technology and Application
{gzli, pwang, kewenjun, jiajliu, keji, ziyus1999, zijiexu}@seu.edu.cn

Abstract

Relation extraction (RE) aims to identify relations between entities mentioned in texts. Although large language models (LLMs) have demonstrated impressive in-context learning (ICL) abilities in various tasks, they still suffer from poor performances compared to most supervised fine-tuned RE methods. Utilizing ICL for RE with LLMs encounters two challenges: (1) retrieving good demonstrations from training examples, and (2) enabling LLMs exhibit strong ICL abilities in RE. On the one hand, retrieving good demonstrations is a non-trivial process in RE, which easily results in low relevance regarding entities and relations. On the other hand, ICL with an LLM achieves poor performance in RE while RE is different from language modeling in nature or the LLM is not large enough. In this work, we propose a novel recall-retrieve-reason RE framework that synergizes LLMs with retrieval corpora (training examples) to enable relevant retrieving and reliable in-context reasoning. Specifically, we distill the consistently ontological knowledge from training datasets to let LLMs generate relevant entity pairs grounded by retrieval corpora as valid queries. These entity pairs are then used to retrieve relevant training examples from the retrieval corpora as demonstrations for LLMs to conduct better ICL via instruction tuning. Extensive experiments on different LLMs and RE datasets demonstrate that our method generates relevant and valid entity pairs and boosts ICL abilities of LLMs, achieving competitive or new state-of-the-art performance on sentence-level RE compared to previous supervised fine-tuning methods and ICL-based methods.

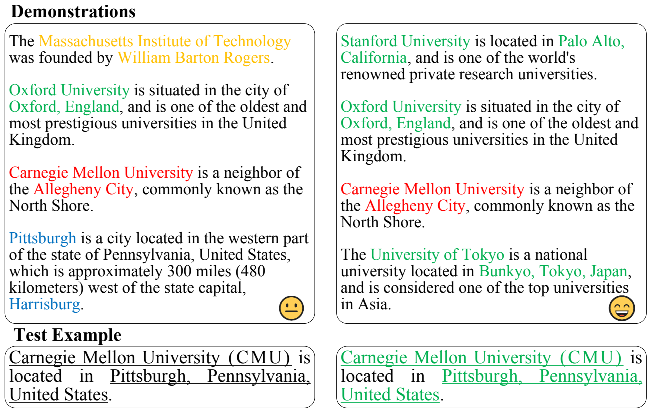
1 Introduction

The emergence of large language models (LLMs) such as GPT-3 [Brown *et al.*, 2020] represent a significant advancement in natural language processing (NLP). Instead of following a pre-training then fine-tuning pipeline [Radford *et al.*, 2019; Devlin *et al.*, 2019; Liu *et al.*, 2019; Raffel *et al.*, 2020], which fine-tunes a pre-trained model on a task-specific dataset in a fully-supervised manner, LLMs employ a new paradigm known as in-context learning (ICL) [Brown *et al.*, 2020] which formulates an NLP task under the paradigm of language generation and makes predictions by learning from a few demonstrations. LLMs with ICL demonstrate impressive performance comparable to traditional methods even with limited examples [Brown *et al.*, 2020; Zhao *et al.*, 2021].

Despite the generally promising ICL performances [Wei *et al.*, 2022b; Arora *et al.*, 2023; Shang *et al.*, 2024], current ICL in relation extraction (RE) [Li *et al.*, 2024] task suffers from relatively poor performance. RE is a pivotal task in NLP [Li *et al.*, 2022; Wang *et al.*, 2023a; Wang *et al.*, 2023b; Ji *et al.*, 2023; Liu *et al.*, 2024], necessitating a profound comprehension of natural language, which involves identifying a pre-defined relation between a given entity pair mentioned in the input sentence or marking it as `NA` if no relation is identified. Given a test example input, ICL for RE prompts the input of LLMs with a few demonstrations retrieved from the training data and the test input itself, then LLMs generate the corresponding relation. Recent studies [Jimenez Gutierrez *et al.*, 2022; Ma *et al.*, 2023] have revealed a significant performance gap in LLMs when apply ICL to the RE task. The main obstacles that utilizing ICL for RE with LLMs are two-fold: (i) the low relevance regarding entity and relation in the retrieved demonstrations for ICL [Wan *et al.*, 2023], and (ii) the failure of utilizing LLMs with moderate size (less than 10B) compared to 175B GPT-3 for ICL [Li *et al.*, 2023c].

Typically, ICL demonstrations are selected randomly or via similarity-based sentence embedding obtained by sentence encoder such as Sentence-BERT [Reimers and Gurevych, 2019]. However, similarity-based retrieval is more concerned with the relevance of the overall sentence semantics and not as much with the specific entities and relations it contains, where consequently the test input tends to retrieve a semantically similar sentence but is not desired in terms of entities and relations. Similar with finding good demonstrations in ICL for general NLP tasks [Liu *et al.*, 2022], the demonstrations for RE should contain as much as similar or exactly same entities and relations regarding the test example, so as to significantly help the test example infer the relation between entities. However, current retrieval techniques [Gao *et al.*, 2021; Wan *et al.*, 2023] fail to locate the delicate demonstrations considering the consistency of ontology (i.e., gen-

[*] Corresponding author



Demonstrations

The Massachusetts Institute of Technology was founded by William Barton Rogers.

Oxford University is situated in the city of Oxford, England, and is one of the oldest and most prestigious universities in the United Kingdom.

Carnegie Mellon University is a neighbor of the Allegheny City, commonly known as the North Shore.

Pittsburgh is a city located in the western part of the state of Pennsylvania, United States, which is approximately 300 miles (480 kilometers) west of the state capital, Harrisburg. 😐

Stanford University is located in Palo Alto, California, and is one of the world's renowned private research universities.

Oxford University is situated in the city of Oxford, England, and is one of the oldest and most prestigious universities in the United Kingdom.

Carnegie Mellon University is a neighbor of the Allegheny City, commonly known as the North Shore.

The University of Tokyo is a national university located in Bunkyo, Tokyo, Japan, and is considered one of the top universities in Asia. 😄

Test Example

Carnegie Mellon University (CMU) is located in Pittsburgh, Pennsylvania, United States.

Carnegie Mellon University (CMU) is located in Pittsburgh, Pennsylvania, United States.

Figure 1: Comparison between naive demonstration selection (left) and our demonstration selection (right) methods. Different colors represent different relations between entity pairs, while green represents the golden relation expressed by entity pairs in test example.

eralized entity types and the relations between them). Since entity types grouped under the similar ontology are often correlated and share similar relations [Roche, 2003], we assume that examples sharing the similar ontology are relevant and could serve as good demonstrations for ICL. To this end, we propose to retrieve the relevant demonstrations in RE incorporating consistent ontological knowledge. Figure 1 illustrates the basic principles of our demonstration selection method, where we expect the demonstrations to include as many entity pairs as possible that express the same relation (i.e., similar ontology) with the test example. To obtain such ontological knowledge guided demonstrations, we propose to let the LLMs learn to generate relevant entity pairs that share the same relations with each test example, by implicitly acquiring the ontological knowledge of specific RE task during training phase. Then ICL demonstrations are retrieved from training examples by these recalled entity pairs. Compared to similarity-based retrieval, recalling entities explicitly via implicitly ontological knowledge guiding bring more benefits.

Besides the demonstration selection, the parameter scale of LLMs also plays an important role in ICL. Existing promising RE results obtaining via ICL all use very large-scale proprietary models [Agrawal *et al.*, 2022; Li *et al.*, 2023c]. Although smaller open-source LLMs such as LLaMA [Touvron *et al.*, 2023] can also perform ICL similarly in simple classification tasks like sentiment classification, they are unable to perform more challenge tasks like extracting relations between entities through in-context demonstrations. Due to the limitations of proprietary models (higher computation cost and time consumption, concerns about privacy protection and local deployment, etc.), open-source smaller LLMs (less than 10B) achieve better ICL performance is of more applicability. Therefore, recent studies [Chen *et al.*, 2022b; Min *et al.*, 2022b] attempt to boost the moderate size LLMs ICL ability via meta in-context learning where an LLM is tuned to do in-context learning on a large set of training tasks. Inspired by this, we optimize LLMs to do in-context RE on training set, making the model becomes more effectively to reason about the relation between entities in-context by conditioning on a few training examples at inference time. Compared to directly optimize the input-output formats of examples (sentence and entity pair as input and relation label as output), this method can conduct in-context reasoning based on retrieved examples and generate more accurate results.

To alleviate the above issues of irrelevant demonstration retrieval and inferior ICL capability, we present a recall-retrieve-reason framework, a novel RE method called **RE**^{4} (**R**elation **E**xtraction with **RE**call, **RE**trieve and **RE**ason) that synergizes open-source LLMs with retrieval corpora (training examples) to retrieve relevant demonstrations and conduct in-context reasoning. Specifically, RE^4 first generates entity pairs guided by ontological knowledge and grounded by retrieval corpora as valid queries via the recalling module. These entity pairs are then used to retrieve valid training examples from retrieval corpora in retrieval module to conduct in-context reasoning by reasoning module. In this way, we not only retrieve the relevant demonstrations from retrieval corpora but also consider the guidance of entity pairs and relations for reasoning. Based on this framework, RE^4 is joint optimized by two tasks: 1) recalling optimization, where we distill ontological knowledge from retrieval corpora into LLMs to generate relevant and valid entity pairs as queries; and 2) reasoning optimization, where we enable LLMs to conduct in-context reasoning based on retrieved demonstrations and generate predicted relations. We conduct extensive experiments on RE benchmarks to validate the effectiveness of RE^4. In summary, our contributions are three-fold:

- We propose a novel recall-retrieve-reason framework for RE that synergizes open-source LLMs with training examples to retrieve relevant demonstrations and conduct in-context reasoning. Moreover, the recalling module of RE^4 can be plug-and-play with different LLMs during inference to improve their performance.
- We distill the consistently ontological knowledge from training examples to guide LLMs for valid and relevant entity generation, and propose to boost the open-source LLMs in-context reasoning for RE via ICL tuning.
- We conduct extensive experiments on RE benchmarks and the results demonstrate that RE^4 achieves state-of-the-art performance in sentence-level RE.

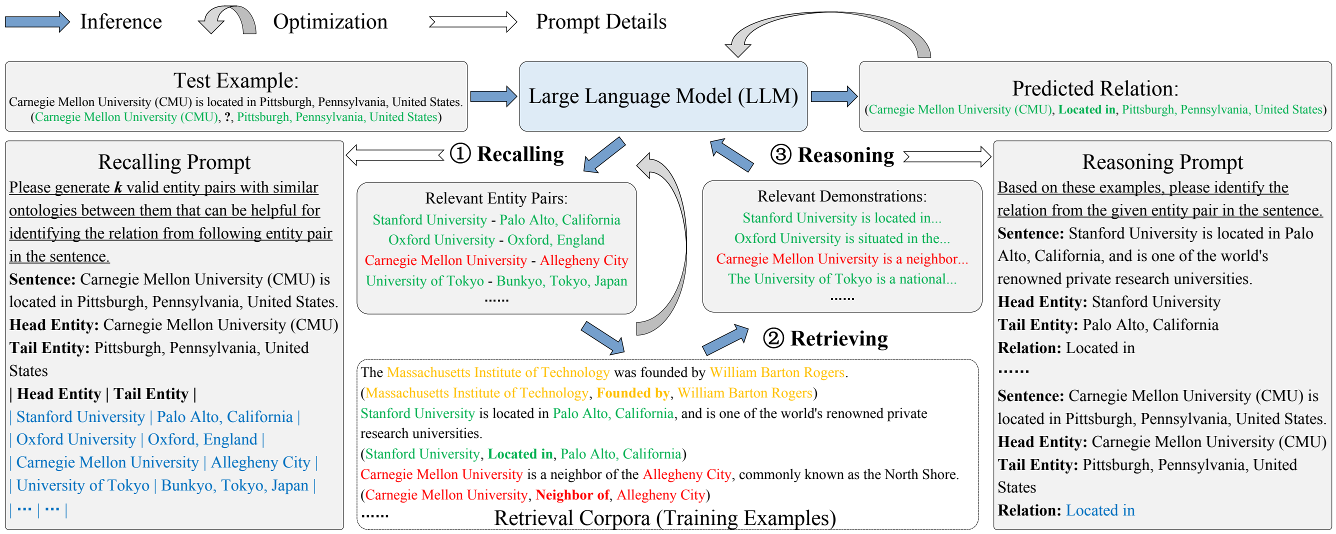
2 Methodology

2.1 Task Formulations

For a given natural language sentence s contains N words, relation extraction aims at extracting the pre-defined relationship r between h and t, where h and t are two given target entities in s. If there is no pre-defined relation between h and t, predict NA. Following previous works [Cabot and Navigli, 2021; Paolini *et al.*, 2021; Li *et al.*, 2023b], we treat RE as a generation task following the standard decoding manner. The optimization procedure can be written as follows:

$$p(r|s,h,t)=\prod_{l=1}^{L}p(y_l|s,h,t,y_{<l}) \tag{1}$$

where we use an auto-regressive transformer [Vaswani *et al.*, 2017] decoder to generate the relation name r with label



Test Example:
Carnegie Mellon University (CMU) is located in Pittsburgh, Pennsylvania, United States.
(Carnegie Mellon University (CMU), **?**, Pittsburgh, Pennsylvania, United States)

Predicted Relation:
(Carnegie Mellon University (CMU), **Located in**, Pittsburgh, Pennsylvania, United States)

Recalling Prompt

Please generate ***k*** valid entity pairs with similar ontologies between them that can be helpful for identifying the relation from following entity pair in the sentence.
Sentence: Carnegie Mellon University (CMU) is located in Pittsburgh, Pennsylvania, United States.
Head Entity: Carnegie Mellon University (CMU)
Tail Entity: Pittsburgh, Pennsylvania, United States
| Head Entity | Tail Entity |
Stanford University	Palo Alto, California
Oxford University	Oxford, England
Carnegie Mellon University	Allegheny City
University of Tokyo	Bunkyo, Tokyo, Japan
···	···

Relevant Entity Pairs:
Stanford University - Palo Alto, California
Oxford University - Oxford, England
Carnegie Mellon University - Allegheny City
University of Tokyo - Bunkyo, Tokyo, Japan
......

Relevant Demonstrations:
Stanford University is located in...
Oxford University is situated in the...
Carnegie Mellon University is a neighbor...
The University of Tokyo is a national...
......

The Massachusetts Institute of Technology was founded by William Barton Rogers.
(Massachusetts Institute of Technology, **Founded by**, William Barton Rogers)
Stanford University is located in Palo Alto, California, and is one of the world's renowned private research universities.
(Stanford University, **Located in**, Palo Alto, California)
Carnegie Mellon University is a neighbor of the Allegheny City, commonly known as the North Shore.
(Carnegie Mellon University, **Neighbor of**, Allegheny City)
......
Retrieval Corpora (Training Examples)

Reasoning Prompt

Based on these examples, please identify the relation from the given entity pair in the sentence.
Sentence: Stanford University is located in Palo Alto, California, and is one of the world's renowned private research universities.
Head Entity: Stanford University
Tail Entity: Palo Alto, California
Relation: Located in
......
Sentence: Carnegie Mellon University (CMU) is located in Pittsburgh, Pennsylvania, United States.
Head Entity: Carnegie Mellon University (CMU)
Tail Entity: Pittsburgh, Pennsylvania, United States
Relation: Located in

Figure 2: Illustration of the RE^4 framework. Given a test example, we first prompt LLMs to generate several relevant entity pairs that are grounded by retrieval corpora as queries. Then we retrieve demonstrations from training examples using the queries. Finally, we conduct in-context reasoning based on the retrieved entities and relations. The instructions in prompts are marked with underline, and the outputs of LLMs in prompts are highlighted in blue.

length L, and $p(y_l|s, h, t, y_{<l})$ denotes the probability of each token y in r generated by LLMs.

2.2 Framework Overview

Recently, many techniques have been explored to improve the ICL ability of LLMs in RE, which first retrieves relevant demonstrations from training examples and then conduct in-context RE based on them [Wan *et al.*, 2023]. However, the challenges of low-quality demonstration retrieval and inferior ICL ability of open-source models hinder the progress of ICL for RE in the era of LLMs. To address these issues, we propose a novel recall-retrieve-reason framework, which recalls relevant entity pairs, retrieves good demonstrations and then perform in-context reasoning for better relation predictions. The overall framework of RE^4 is illustrated in Figure 2. Given a test example "*Carnegie Mellon University (CMU) is located in Pittsburgh, Pennsylvania, United States*", aiming to identify the relation between "*Carnegie Mellon University (CMU)*" and "*Pittsburgh, Pennsylvania, United States*", we generate an entity pair "*Stanford University*" and "*Palo Alto, California*" as the query. This entity pair expresses the similar relation with the test entity pair as the consistency of ontology between them (i.e., the head entity type is *Organization* and the tail entity type is *Location*). Then we retrieve the corresponding demonstration from training examples. Finally, we predict the relation (i.e. *Located in*) based on demonstrations.

2.3 Framework Optimization

We formulate our RE^4 as an optimization problem that aims to maximize the probability of reasoning the relation r from a retrieval corpora $\mathcal{C}$ w.r.t the test example e by generating consistently ontological entity pairs z as the queries:

$$p_\theta(r|e, \mathcal{C}) = \sum_{z \in \mathcal{Z}} p_\theta(r|e, z, \mathcal{C}) p_\theta(z|e) \tag{2}$$

where θ denotes the parameters of LLMs, z denotes the entity pairs (queries) generated by LLMs, and $\mathcal{Z}$ denotes the set of entity pairs that share similar ontology with the test example e. The latter term $p_\theta(z|e)$ is the probability of generating a valid entity pair z grounded by retrieval corpora given e, which is realized by the recalling module. The former term $p_\theta(r|e, z, \mathcal{C})$ is the probability of reasoning the relation r in context given the test example e, entity pair z, and retrieval corpora $\mathcal{C}$, computing by the reasoning module.

Despite the advantage of generating entity pairs as queries, the LLMs have zero ontological knowledge of the entities and relations contained in training examples. Therefore, LLMs cannot directly generate entity pairs grounded by retrieval corpora as valid queries. Moreover, LLMs might not have strong ICL ability to conduct effective in-context reasoning based on them. To address these issues, we design two instruction tuning tasks: 1) recalling optimization, which distills the consistent ontological knowledge from training examples into LLMs to generate valid entity pairs as queries, and 2) reasoning optimization, which enables LLMs to perform in-context reasoning based on retrieved demonstrations. The objective function in equation (2) is optimized by maximizing the evidence lower bound, which is formulated as:

$$\begin{aligned} \log p_\theta(r|e, \mathcal{C}) \geq\ & \mathbb{E}_{z \sim q(z)}[\log p_\theta(r|e, z, \mathcal{C})] \\ & - D_{\mathrm{KL}}(q(z) \,||\, p_\theta(z|e)) \end{aligned} \tag{3}$$

where $q(z)$ denotes the posterior distribution of valid and relevant entity pairs grounded by retrieval corpora. The latter term minimizes the KL divergence between the posterior and the prior, which encourages LLMs to generate consistently ontological entity pairs (i.e. recalling optimization). The former term maximizes the expectation that reasoning module generates correct relations based on the retrieved demonstrations (i.e. reasoning optimization).

2.4 Recalling Entity Pairs

To make LLMs generate valid entity pairs as queries for retrieving consistently ontological demonstrations from retrieval corpora, we minimize the KL divergence with the posterior distribution of valid entity pairs $q(z)$, which can be approximated by the valid entity pairs in retrieval corpora $\mathcal{C}$.

Given a test example e and its golden relation r, we could find the entity pair instances $z = (h, t)$ expressing the same relation r in training examples. And z can be considered valid and serve as a query for retrieving the relevant demonstration of e. And $q(z)$ can be formally approximated as:

$$q(z) \cong q(z|r, e, \mathcal{C}) = \frac{1}{|\mathcal{Z}|}, \exists\, z \in \mathcal{C} \tag{4}$$

where we assume a uniform distribution over all consistently ontological entity pairs $\mathcal{Z}$ regarding e, and $\exists\, z \in \mathcal{C}$ denotes the existence of an entity pair instance connecting e and r in $\mathcal{C}$. Therefore, the KL divergence can be calculated as:

$$\mathcal{L}_{\text{recall}} = D_{\text{KL}}(q(z)\,||\,p_\theta(z|e)) \cong -\frac{1}{|\mathcal{Z}^*|}\sum_{z\in\mathcal{Z}^*}\log p_\theta(z|e) \tag{5}$$

where we use the partial entity pairs $\mathcal{Z}^* \subset \mathcal{Z}$ relevant to e in retrieval corpora $\mathcal{C}$ as supervision signals [Luo *et al.*, 2023], where we distill the consistently ontological knowledge from training examples to LLMs. To utilize the instruction-following ability of LLMs [Wei *et al.*, 2022a], we design an instruction template that prompts LLMs to generate $k = |\mathcal{Z}^*|$ entity pairs. Therefore, the optimization of $\mathcal{L}_{\text{recall}}$ becomes:

$$-\frac{1}{|\mathcal{Z}^*|}\sum_{z\in\mathcal{Z}^*}\log p_\theta(z|e) = -\frac{1}{|\mathcal{Z}^*|}\sum_{z\in\mathcal{Z}^*}\log\prod_{i=1}^{|z|}p_\theta(y_i|y_{<i}, e) \tag{6}$$

where $p_\theta(z|e)$ denotes the prior distribution of generating valid entity pair z, and $p_\theta(y_i|y_{<i}, e)$ denotes the probability of each token in z generated by LLMs.

2.5 Retrieving From Corpora

Given a test example e and an entity pair as query z, the retrieving module aims to retrieve the relevant demonstration d from retrieval corpora. The retrieval process can be simply conducted by exact match between the generated entity pair z and the entity pair of each candidate demonstration d:

$$\mathcal{D} = \{d\,|\,h_z = h_d, t_z = t_d, z = (h_z, t_z), d = (s, h_d, t_d)\} \tag{7}$$

where $\mathcal{D}$ denotes the set of retrieved demonstrations for ICL. Despite we can utilize the retrieved entity pairs to directly get the predicted relations via majority vote, the retrieved entity pairs of relations could be noisy and imperfect to the test example e, leading to incorrect predictions. Therefore, we use a reasoning module to boost the ICL ability of LLMs to identity the important entity pairs of relations and predict relations based on them via in-context reasoning [Chen *et al.*, 2023].

2.6 Reasoning Through Demonstrations

In reasoning module, we aim to enable LLMs to conduct in-context reasoning based on the relevant entity pairs. And reasoning on multiple relevant entity pairs is formulated as:

$$p_\theta(r|e, \mathcal{Z}, \mathcal{C}) = \prod_{z\in\mathcal{Z}} p_\theta(r|e, z, \mathcal{C}) \tag{8}$$

By approximating the expectation with k sampled entity pairs $\mathcal{Z}^*$, the objective function of reasoning optimization where maximizes the probability of LLMs generating golden relations based on the relevant entity pairs is formalized as :

$$\begin{aligned}\mathcal{L}_{\text{reason}} &= -\mathbb{E}_{z\sim q(z)}[\log p_\theta(r|e, z, \mathcal{C})] \\ &= -\sum_{z\in\mathcal{Z}^*}\log p_\theta(r|e, z, \mathcal{C}) \\ &= -\log p_\theta(r|e, \mathcal{Z}^*, \mathcal{C})\end{aligned} \tag{9}$$

The reasoning module takes the test example e and a set of retrieved demonstrations $\mathcal{D}$ to generate relation r. Similar with recalling module, we design a reasoning instruction prompt to guide LLMs to conduct in-context reasoning based on the retrieved demonstrations $\mathcal{D}$. The $\mathcal{D}$ are formulated as a series of structural sentences in standard ICL paradigm. The optimization of $\mathcal{L}_{\text{reason}}$ is formulated as:

$$\log p_\theta(r|e, \mathcal{Z}^*, \mathcal{C}) = \log\sum_{z\in\mathcal{Z}^*}\sum_{d\in\mathcal{D}}\prod_{i=1}^{|r|}p_\theta(y_i|y_{<i}, e, d) \tag{10}$$

where $p_\theta(r|e, \mathcal{Z}^*, \mathcal{C})$ denotes probability of reasoning the golden relation r based on k retrieved demonstrations $\mathcal{Z}^*$, and y_i denotes the i-th token of relation r. To reduce the impact of error propagation during recalling process and improve the robustness of the model for in-context reasoning during inference time, we add some noise into $\mathcal{Z}^*$ by replacing $k^*, (1 \le k^* \le k)$ of k demonstrations in $\mathcal{Z}^*$ with training examples share different relations from retrieval corpora with a uniform distribution. In this way, the LLMs can learn to in-context reasoning [Chen *et al.*, 2022b; Min *et al.*, 2022b; Coda-Forno *et al.*, 2023] based on important entities and relations, avoiding simply deduce the relations via majority vote.

2.7 Joint Optimization and Inference

The final objective function of RE^4 is the combination of the recalling optimization and reasoning optimization, which can be formulated as:

$$\begin{aligned}\mathcal{L} = &-\frac{1}{|\mathcal{Z}^*|}\sum_{z\in\mathcal{Z}^*}\log\prod_{i=1}^{|z|}p_\theta(y_i|y_{<i}, e) \\ &-\log\sum_{z\in\mathcal{Z}^*}\sum_{d\in\mathcal{D}}\prod_{i=1}^{|r|}p_\theta(y_i|y_{<i}, e, d)\end{aligned} \tag{11}$$

We adopt the same LLM for both recalling and reasoning, which are jointly trained on two instruction tuning tasks. To enhance the efficiency of the fine-tuning process and reduce memory requirements, we utilize Low-Rank Adaptation (LoRA) [Hu *et al.*, 2022], which freezes the pre-trained model weights and injects trainable rank decomposition matrices into each layer of the Transformer [Vaswani *et al.*, 2017] architecture, greatly reducing the number of trainable parameters for downstream tasks. During inference, based on the recalled entity pairs and retrieved demonstrations, the LLMs conduct ICL to generate predicted relations.

3 Experiments

3.1 Settings

Datasets and Metrics. We evaluate RE^4 on SemEval 2010 [Hendrickx *et al.*, 2019], TACRED [Zhang *et al.*, 2017], Google RE [1], SciERC [Luan *et al.*, 2018], four commonly used RE datasets [2]. Following previous works [Yamada *et al.*, 2020; Cabot and Navigli, 2021; Li *et al.*, 2023a], we use the Micro-F1 score excluding `NA` as the metric for evaluation. The statistics of datasets are shown in Table 1.

Dataset	#Relation	#Train	#Dev	#Test
SemEval	9	6,507	1,493	2,717
TACRED	41	68,124	22,631	15,509
Google RE	5	38,112	9,648	9,616
SciERC	7	3,219	455	974

Table 1: Statistics of datasets.

Baselines. We compare RE^4 with state-of-the-art RE models that represent a diverse array of approaches. Supervised fine-tuning (SFT) RE methods can be divided into three categories. Classification-based methods fine-tune language models on RE datasets with classification losses, such as MTB [Baldini Soares *et al.*, 2019], LUKE [Yamada *et al.*, 2020], IRE [Zhou and Chen, 2022] and KLG [Li *et al.*, 2023a]. Prompt-based methods use prompt and treats RE as a cloze-style task, such as KnowPrompt [Chen *et al.*, 2022a] and NLI-DeBERTa [Sainz *et al.*, 2021]. Generative-based methods use text generation models for RE, such as REBEL [Cabot and Navigli, 2021], TANL [Paolini *et al.*, 2021], RELA [Li *et al.*, 2023b] and DeepStruct [Wang *et al.*, 2022]. For ICL-based RE, existing methods typically rely on the strong ICL ability of large-scale proprietary models without any SFT. We utilize GPT-RE [Wan *et al.*, 2023] which use PURE [Zhong and Chen, 2021] as the demonstration retriever and GPT-3 [Brown *et al.*, 2020] as the base LLM for ICL. RE^4 can be regarded as the combination of SFT and ICL paradigms. On the one hand, it makes LLMs better adapt to specific tasks via instruction tuning. On the other hand, it improves the in-context reasoning ability of LLMs and enables relevant retrieved demonstrations during the ICL process.

Experiment Details. We experiment RE^4 with open-source LLMs including T5 [Raffel *et al.*, 2020], BART [Lewis *et al.*, 2020] and LLaMA [Touvron *et al.*, 2023]. For model scales, we select T5-Base (220M), T5-Large (770M), BART-Base (140M), BART-Large (400M) and LLaMA-7B for experiments. We utilize LoRA [Hu *et al.*, 2022] to tune LLMs for simplicity and efficiency. We set the rank r of the LoRA parameters to 8 and the merging ratio α to 32. We train RE^4 for 5 epochs with batch size 4 and learning rate 1e-4. For the number of generated entity pairs, we set k to 5. The checkpoint of LoRA adapter that achieves the best result on the validation set is used for testing. We also directly treat relation names as generation objectives and fine-tune the LLaMA.

Method (#Param.)	**SemEval**	**TACRED**	**Google RE**	**SciERC**
MTB (336M)	89.5	71.5	92.7	87.4
LUKE (355M)	90.1	72.7	94.0	87.7
IRE (355M)	89.8	74.6	93.1	88.9
KLG (355M)	90.5	75.6	-	-
KnowPrompt (355M)	90.2	72.4	-	-
NLI-DeBERTa (1.5B)	-	73.9	-	-
REBEL (400M)	-	73.7	93.5	86.3
TANL (220M)	-	74.8	-	-
RELA (400M)	90.4	71.2	93.9	90.3
DeepStruct (10B)	-	76.8	-	-
GPT-3 (175B)	70.1	32.5	-	-
GPT-RE (175B)	91.9	72.1	-	-
BART-Base w/ RE^4	89.8	71.5	92.4	86.0
BART-Large w/ RE^4	90.6	73.3	93.1	87.2
T5-Base w/ RE^4	89.9	72.7	92.6	86.3
T5-Large w/ RE^4	90.9	75.6	93.4	87.8
LLaMA w/ RE^4	**92.1**	**77.2**	**94.5**	**91.7**
LLaMA w/o RE^4	90.6	75.0	92.9	89.5

Table 2: Micro-F1 score of test sets on four RE datasets. Results of baselines are retrieved from original papers. For ICL-based results, we use best 30-shot on SemEval and 15-shot on TACRED. Previous state-of-the-art results are marked with underline, and best results are **bold**. Results of RE^4 are averaged over three random seeds.

3.2 Main Results

The main results of baselines and RE^4 are summarized in Table 2, where RE^4 with LLaMA outperforms all previous state-of-the-art methods on four RE datasets. Compared to the classification-based methods, RE^4 shows favorable results without any external dataset usage or additional pre-training stages, while MTB and LUKE all involve entity and relation related pre-training tasks. RE^4 with most LLMs also suppresses two prompt-based methods. Compared to previous generative-based methods, RE^4 consistently achieves competitive or superior results on four datasets. For example, with same backbone BART-Large, RE^4 and REBEL share similar overall results, while REBEL is pre-trained with a large external relational triple extraction dataset. Notably, RE^4 with LLaMA significantly outperforms REBEL on TACRED and SciERC, and vanilla fine-tuned LLaMA still surpasses REBEL, due to the fact that LLaMA (7B) has a much larger number of model parameters than REBEL (400M). Although LLaMA is smaller than DeepStruct which is based on a pre-trained 10B parameter language model GLM [Du *et al.*, 2022] and is pre-trained on a collection of large-scale corpus, RE^4 delivers better results on TACRED than DeepStruct. This is notable because RE^4 adopt smaller foundation models for fine-tuning and utilize no external datasets for pre-training compared to DeepStruct. In addition, RE^4 with LLaMA achieves comparable results on SemEval and much better results on TACRED compared to ICL-based method GPT-RE. Without fine-tuning, if GPT-3 lacks relevant domain knowledge about specific tasks, then the performance of GPT-RE is limited and greatly affected by the retrieved demonstrations. And this is why GPT-RE delivers exceptional re-

[1] https://github.com/google-research-datasets/relation-extraction-corpus

[2] https://github.com/1iguozheng/RE4

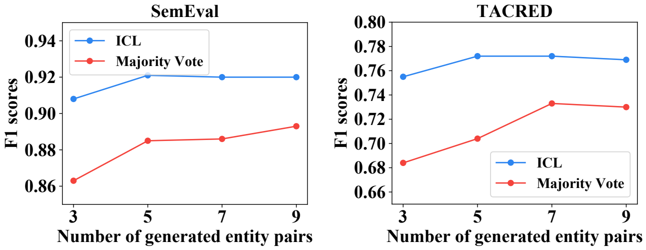


Figure 3: The sensitivity of k. **ICL** denotes we perform in-context reasoning in reasoning module with retrieved demonstrations, while **Majority Vote** denotes we consider the relation with the maximum number of generated entity pairs as the predicted relation.

Dataset	#Test	#Entity Pairs	#Valid	#Ratio
SemEval	2,717	13,585	13,023	95.86%
TACRED	15,509	77,545	73,482	94.76%

Table 3: Validness of generated entity pairs (k=5). #Valid denotes the number of valid generated entity pairs, and #Ratio denotes the ratio of valid entity pairs in all generated entity pairs.

sults on SemEval but unattractive results that even worse than LUKE on TACRED. In other words, solely relying on ICL-based methods is unstable. The main results demonstrate the simplicity and effectiveness of RE^4 compared to baselines.

3.3 Discussions

We conduct extensive experiments to verify the effectiveness of recalling and reasoning module. We also provide detailed analysis about retrieved demonstrations and in-context reasoning abilities. In this section, we only experiment RE^4 with LLaMA on SemEval and TACRED for simplicity.

Number of Generated Entity Pairs. We vary the number of generated entity pairs k from 3, 5, 7 to 9. We also consider the majority vote results, the overall results are shown in Figure 3. On the one hand, we find that without in-context reasoning, the performance are more sensitive to the k-selection. With the help of ICL, RE^4 could achieve more stable and much better performances, which highlights the importance of reasoning module. On the other hand, obviously, setting k to 5 delivers the best results across two datasets. While increasing k generally boosts the performance of majority vote, the overall performance of RE^4 tends to stabilize, even with a slight decrease. This is because in-context reasoning is sensitive to the distractors (i.e., additional retrieved demonstrations that are not relevant to a test example) [Shi *et al.*, 2023]. Therefore, setting larger k not only increases difficulty in generating accurately consistently ontological entities, but brings more distractors that are negative for reasoning module.

Quality of Generated Entity Pairs. We consider two aspects of generated entity pairs for quality checking. We first examine the validness of generated entity pairs (i.e., whether it is grounded by retrieval corpora), then we evaluate the relevance of retrieved demonstrations with respect to test examples. We match the generated entity pairs with the entities in training examples, the results are shown in Table 3. It can be seen that under strict exact matching, the proportion of valid generated entity pairs still reaches around 95%, which indicates that the entity pairs generated by recalling module could serve as the faithful queries for retrieval corpora. However, we are still interested in those invalid entity pairs. Take the SemEval dataset as the example, we analyze the remaining 4.14% invalid entity pairs, discovering that 70 of 562 entity pairs at least one head or tail entity is grounded by retrieval corpora. The other 492 generated entity pairs all have valid entities, but cannot express a valid relation. Overall, during entity pairs generation, the recalling module reduces the hallucination issue of the LLMs and generates faithful queries.

Dataset	5	4	3	2	1	0
SemEval	71.03%	7.45%	6.49%	5.45%	5.21%	4.37%
TACRED	51.88%	12.75%	5.20%	9.74%	12.26%	8.17%

Table 4: The number of retrieved examples which share same relation with a test example (k = 5). We consider all the examples in test sets and calculate the proportion.

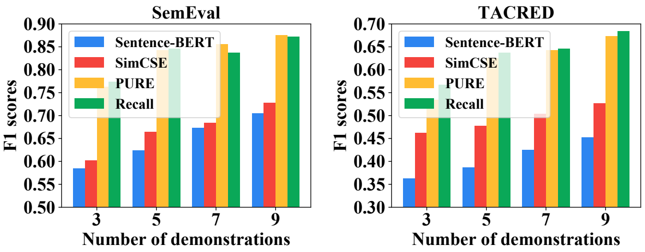


Figure 4: Comparison on different retrieval models.

To evaluate the relevance of retrieved demonstrations with respect to test examples, we consider the number of examples which share same relation with a test example (i.e. the consistency of ontology). The results are summarized in Table 4. For most cases, the entity pairs generated by the recalling module and the demonstrations retrieved by retrieving module are all relevant (i.e., all 5 demonstrations share same relation with a test example). Note that 84.97% and 69.83% of retrieved results ensure the majority of golden relation examples participation. And these demonstrations can provide relevant contextual knowledge for ICL. However, one drawback of recalling is that there may be no golden relation in the generated results. Existing similarity-based retrievers such as Sentence-BERT [Reimers and Gurevych, 2019], SimCSE [Gao *et al.*, 2021] and PURE [Zhong and Chen, 2021] essentially cannot guarantee that the retrieved demonstrations would always contain the golden relation. The relevance of retrieved demonstrations actually rely on the performance of the retrieval model. Therefore, we evaluate the quality of retrieved demonstrations from recalling module and similarity calculation via ICL on GPT-3 [3], as shown in Figure 4. We find that fine-tuned retrievers (PURE and recalling module)

[3] For GPT-3, we use "`text-davinci-003`". For Sentence-BERT, we use "`all-mpnet-base-v2`". For SimCSE, we use "`sup-simcse-bert-base-uncased`". For PURE, we use "`bert-base-uncased`" as the backbone.

Training	Inference	SemEval			TACRED		
		I	**II**	**Avg.**	**I**	**II**	**Avg.**
w/o tuning	Direct / ICL	0.0	0.0	0.0	0.0	0.0	0.0
w/ fine-tuning	Direct	90.7	**90.1**	90.4	74.6	**75.2**	74.9
w/ ICL tuning	ICL	**91.4**	89.8	**90.6**	**76.7**	73.7	**75.2**

Table 5: Comparison between vanilla fine-tuning and ICL tuning. **I** denotes the model performance on test examples that their demonstrations contain at least one golden relation example, while **II** denotes the performance when the random selected demonstrations are all distractors. **Avg.** represents the average score of **I** and **II** results.

substantially achieve much better ICL results than similarity-based models without considering entity and relation semantics. For different number of demonstrations, recalling module delivers comparable results compared to PURE, showing the effectiveness of recalling module and allowing seamless integration with any arbitrary LLMs during inference.

Effectiveness of ICL Tuning. We remove the recalling instruction task and keep the reasoning instruction task. Not relying on the retrieved demonstrations $\mathcal{Z}^*$, we use 5 training examples as demonstrations, sampled uniformly at random, during both training and testing time. Note that we relax the assumption of perfect balance between labels on training examples. We categorize two types of test examples and then compare the performance of RE^4 with vanilla fine-tuned LLaMA, the results are shown in Table 5. First, without any tuning process, LLaMA cannot perform ICL in RE. We empirically discover that LLaMA is unable to understand the structural sentences in standard ICL paradigm and recover the relation labels of test examples based on demonstrations. Second, although the distractors can lead LLMs to make inaccurate predictions, the improvement brought by ICL tuning is still obvious. Experimental results suggest that the impact of ICL training is positive when the golden relation is included in retrieved demonstrations compared to vanilla fine-tuning, which also highlights the importance and performance of retrieval models. During instruction tuning, the model learns to learn in-context for deducing the golden relation.

Sensitivity of In-Context Reasoning. We have validated that relevant contextual knowledge for test examples is beneficial for overall performance. But we might be interested in whether all retrieved demonstrations share similar ontology are equal during ICL. Specifically, for each test example, we replace the corresponding k generated entity pairs with other k random entity pairs but share same relations. The in-context reasoning results are shown in Figure 5. Similar with other LLMs such as GPT-3, the change of demonstrations also have an impact on the final results. Compared to demonstrations obtained via generated entity pairs, the replacement operation consistently brings slightly performance drop across different k and datasets, which indicates that the recalling module actually help to discover similar demonstrations with test examples. However, the performance degradation caused by this randomness is relatively small compared to performance in GPT-3 [Zhao *et al.*, 2021]. The ICL tuning process forces the LLMs to deduce the golden relation based on important entities and relations, which indicates that RE^4 is able to perform modestly robust reasoning utilizing relevant but imperfect contextual knowledge.

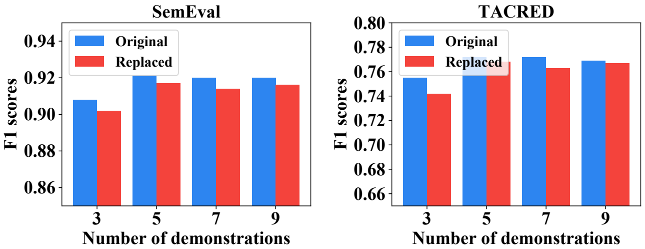


Figure 5: Sensitivity of in-context reasoning.

4 Related Work

The large language models (LLMs), such as GPT-3 [Brown *et al.*, 2020] and ChatGPT [OpenAI, 2022], perform well in various downstream tasks without any training or fine-tuning but only with a few examples as instructions, which is called in-context learning (ICL). However, ICL with LLMs achieves poor performance in relation extraction (RE) where the main obstacles are two-fold: (1) the low relevance regarding entity and relation in the retrieved demonstrations for ICL, and (2) the failure of utilizing LLMs with moderate size for ICL. For the first challenge, existing attempts rely on sentence embedding in retrieval, including the sentence encoders such as Sentence-BERT [Reimers and Gurevych, 2019] and SimCSE [Gao *et al.*, 2021]. Considering entity and relation semantics, GPT-RE [Wan *et al.*, 2023] fine-tunes PURE [Zhong and Chen, 2021] to provide more RE-specific and robust representations for retrieval. For the second challenge, although ICL has been further improved by later work [Zhao *et al.*, 2021; Holtzman *et al.*, 2021; Min *et al.*, 2022a] and shows promising results on a variety of tasks, these researches mainly focus on GPT-3. To improve the ICL ability of other LLMs, current methods propose to make LLMs perform better ICL via meta learning [Chen *et al.*, 2022b] and multi-task learning [Min *et al.*, 2022b]. Inspired by these, our method can be viewed as the integration framework of retrieving and reasoning by instruction tuning. Guided by ontological knowledge, RE^4 generates some possible similar entities and relations, and then performs reliable in-context reasoning based on the retrieved demonstrations.

5 Conclusion

In this work, we propose RE^4, a novel recall-retrieve-reason RE framework for open-source LLMs, which recalls relevant entity pairs, retrieves good demonstrations and then perform better in-context reasoning for RE. We consider distilling consistently ontological knowledge to guide the demonstration retrieval, and tuning LLMs with ICL objective to perform reliable in-context reasoning. Specially, RE^4 allows seamless integration with any arbitrary LLMs during inference. Empirical results show that RE^4 achieves new state-of-the-art sentence-level RE performance in four RE benchmarks. We also demonstrate its effectiveness with extensive experiments, and discuss its advantages and limitations, encouraging more effective generative-based RE methods in the future research.

Acknowledgments

We thank the reviewers for their insightful comments. This work was supported by National Science Foundation of China (Grant Nos.62376057) and the Start-up Research Fund of Southeast University (RF1028623234). All opinions are of the authors and do not reflect the view of sponsors.

References

[Agrawal *et al.*, 2022] Monica Agrawal, Stefan Hegselmann, Hunter Lang, et al. Large language models are few-shot clinical information extractors. In *EMNLP*, 2022.

[Arora *et al.*, 2023] Simran Arora, Avanika Narayan, Mayee F Chen, et al. Ask me anything: A simple strategy for prompting language models. In *ICLR*, 2023.

[Baldini Soares *et al.*, 2019] Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, et al. Matching the blanks: Distributional similarity for relation learning. In *ACL*, 2019.

[Brown *et al.*, 2020] Tom Brown, Benjamin Mann, Nick Ryder, et al. Language models are few-shot learners. In *NeurIPS*, 2020.

[Cabot and Navigli, 2021] Pere-Lluís Huguet Cabot and Roberto Navigli. Rebel: Relation extraction by end-to-end language generation. In *Findings of EMNLP*, 2021.

[Chen *et al.*, 2022a] Xiang Chen, Ningyu Zhang, Xin Xie, et al. Knowprompt: Knowledge-aware prompt-tuning with synergistic optimization for relation extraction. In *WWW*, 2022.

[Chen *et al.*, 2022b] Yanda Chen, Ruiqi Zhong, Sheng Zha, et al. Meta-learning via language model in-context tuning. In *ACL*, 2022.

[Chen *et al.*, 2023] Zeming Chen, Gail Weiss, Eric Mitchell, et al. Reckoning: Reasoning through dynamic knowledge encoding. In *NeurIPS*, 2023.

[Coda-Forno *et al.*, 2023] Julian Coda-Forno, Marcel Binz, Zeynep Akata, et al. Meta-in-context learning in large language models. In *NeurIPS*, 2023.

[Devlin *et al.*, 2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, et al. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019.

[Du *et al.*, 2022] Zhengxiao Du, Yujie Qian, Xiao Liu, et al. GLM: General language model pretraining with autoregressive blank infilling. In *ACL*, 2022.

[Gao *et al.*, 2021] Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings. In *EMNLP*, 2021.

[Hendrickx *et al.*, 2019] Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, et al. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. *arXiv preprint arXiv:1911.10422*, 2019.

[Holtzman *et al.*, 2021] Ari Holtzman, Peter West, Vered Shwartz, et al. Surface form competition: Why the highest probability answer isn't always right. In *EMNLP*, 2021.

[Hu *et al.*, 2022] Edward J Hu, Yelong Shen, Phillip Wallis, et al. Lora: Low-rank adaptation of large language models. In *ICLR*, 2022.

[Ji *et al.*, 2023] Ke Ji, Yixin Lian, Jingsheng Gao, and Baoyuan Wang. Hierarchical verbalizer for few-shot hierarchical text classification. In *ACL*, 2023.

[Jimenez Gutierrez *et al.*, 2022] Bernal Jimenez Gutierrez, Nikolas McNeal, Clayton Washington, et al. Thinking about GPT-3 in-context learning for biomedical IE? think again. In *Findings of EMNLP*, 2022.

[Lewis *et al.*, 2020] Mike Lewis, Yinhan Liu, Naman Goyal, et al. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *ACL*, 2020.

[Li *et al.*, 2022] Guozheng Li, Xu Chen, Peng Wang, et al. Fastre: Towards fast relation extraction with convolutional encoder and improved cascade binary tagging framework. In *IJCAI*, 2022.

[Li *et al.*, 2023a] Bo Li, Wei Ye, Jinglei Zhang, et al. Reviewing labels: Label graph network with top-k prediction set for relation extraction. In *AAAI*, 2023.

[Li *et al.*, 2023b] Bo Li, Dingyao Yu, Wei Ye, et al. Sequence generation with label augmentation for relation extraction. In *AAAI*, 2023.

[Li *et al.*, 2023c] Guozheng Li, Peng Wang, and Wenjun Ke. Revisiting large language models as zero-shot relation extractors. In *Findings of EMNLP*, 2023.

[Li *et al.*, 2024] Guozheng Li, Wenjun Ke, Peng Wang, Zijie Xu, Ke Ji, Jiajun Liu, Ziyu Shang, and Qiqing Luo. Unlocking instructive in-context learning with tabular prompting for relational triple extraction. *arXiv preprint arXiv:2402.13741*, 2024.

[Liu *et al.*, 2019] Yinhan Liu, Myle Ott, Naman Goyal, et al. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[Liu *et al.*, 2022] Jiachang Liu, Dinghan Shen, Yizhe Zhang, et al. What makes good in-context examples for GPT-3? In *DeeLIO*, 2022.

[Liu *et al.*, 2024] Jiajun Liu, Wenjun Ke, Peng Wang, Ziyu Shang, Jinhua Gao, Guozheng Li, Ke Ji, and Yanhe Liu. Towards continual knowledge graph embedding via incremental distillation. In *AAAI*, 2024.

[Luan *et al.*, 2018] Yi Luan, Luheng He, Mari Ostendorf, et al. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. *arXiv preprint arXiv:1808.09602*, 2018.

[Luo *et al.*, 2023] Linhao Luo, Yuan-Fang Li, Gholamreza Haffari, et al. Reasoning on graphs: Faithful and interpretable large language model reasoning. *arXiv preprint arXiv:2310.01061*, 2023.

[Ma *et al.*, 2023] Yubo Ma, Yixin Cao, YongChing Hong, et al. Large language model is not a good few-shot information extractor, but a good reranker for hard samples! In *Findings of EMNLP*, 2023.

[Min *et al.*, 2022a] Sewon Min, Mike Lewis, Hannaneh Hajishirzi, et al. Noisy channel language model prompting for few-shot text classification. In *ACL*, 2022.

[Min *et al.*, 2022b] Sewon Min, Mike Lewis, Luke Zettlemoyer, et al. Metaicl: Learning to learn in context. In *NAACL-HLT*, 2022.

[OpenAI, 2022] OpenAI. Introducing chatgpt, 2022.

[Paolini *et al.*, 2021] Giovanni Paolini, Ben Athiwaratkun, Jason Krone, et al. Structured prediction as translation between augmented natural languages. In *ICLR*, 2021.

[Radford *et al.*, 2019] Alec Radford, Jeffrey Wu, Rewon Child, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[Raffel *et al.*, 2020] Colin Raffel, Noam Shazeer, Adam Roberts, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*, 21(1):5485–5551, 2020.

[Reimers and Gurevych, 2019] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *EMNLP*, 2019.

[Roche, 2003] Christophe Roche. Ontology: a survey. *IFAC Proceedings Volumes*, 36(22):187–192, 2003.

[Sainz *et al.*, 2021] Oscar Sainz, Oier Lopez de Lacalle, Gorka Labaka, et al. Label verbalization and entailment for effective zero-and few-shot relation extraction. In *EMNLP*, 2021.

[Shang *et al.*, 2024] Ziyu Shang, Wenjun Ke, Nana Xiu, Peng Wang, Jiajun Liu, Yanhui Li, Zhizhao Luo, and Ke Ji. Ontofact: Unveiling fantastic fact-skeleton of llms via ontology-driven reinforcement learning. In *AAAI*, 2024.

[Shi *et al.*, 2023] Freda Shi, Xinyun Chen, Kanishka Misra, et al. Large language models can be easily distracted by irrelevant context. In *ICML*, 2023.

[Touvron *et al.*, 2023] Hugo Touvron, Thibaut Lavril, Gautier Izacard, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, et al. Attention is all you need. In *NIPS*, 2017.

[Wan *et al.*, 2023] Zhen Wan, Fei Cheng, Zhuoyuan Mao, et al. GPT-RE: In-context learning for relation extraction using large language models. In *EMNLP*, 2023.

[Wang *et al.*, 2022] Chenguang Wang, Xiao Liu, Zui Chen, et al. Deepstruct: Pretraining of language models for structure prediction. In *Findings of ACL*, 2022.

[Wang *et al.*, 2023a] Peng Wang, Tong Shao, Ke Ji, Guozheng Li, and Wenjun Ke. fmlre: a low-resource relation extraction model based on feature mapping similarity calculation. In *AAAI*, 2023.

[Wang *et al.*, 2023b] Peng Wang, Jiafeng Xie, Xiye Chen, et al. Pascore: a chinese overlapping relation extraction model based on global pointer annotation strategy. In *IJCAI*, 2023.

[Wei *et al.*, 2022a] Jason Wei, Maarten Bosma, Vincent Y Zhao, et al. Finetuned language models are zero-shot learners. In *ICLR*, 2022.

[Wei *et al.*, 2022b] Jason Wei, Xuezhi Wang, Dale Schuurmans, et al. Chain of thought prompting elicits reasoning in large language models. In *NeurIPS*, 2022.

[Yamada *et al.*, 2020] Ikuya Yamada, Akari Asai, Hiroyuki Shindo, et al. Luke: deep contextualized entity representations with entity-aware self-attention. In *EMNLP*, 2020.

[Zhang *et al.*, 2017] Yuhao Zhang, Victor Zhong, Danqi Chen, et al. Position-aware attention and supervised data improve slot filling. In *EMNLP*, 2017.

[Zhao *et al.*, 2021] Zihao Zhao, Eric Wallace, Shi Feng, et al. Calibrate before use: Improving few-shot performance of language models. In *ICML*, 2021.

[Zhong and Chen, 2021] Zexuan Zhong and Danqi Chen. A frustratingly easy approach for entity and relation extraction. In *NAACL-HLT*, 2021.

[Zhou and Chen, 2022] Wenxuan Zhou and Muhao Chen. An improved baseline for sentence-level relation extraction. In *AACL*, 2022.