Unlocking Instructive In-Context Learning with Tabular Prompting for Relational Triple Extraction

Guozheng Li[♣], Wenjun Ke[♣♠], Peng Wang[♣♠(✉)], Zijie Xu[♣]
Ke Ji[♣], Jiajun Liu[♣], Ziyu Shang[♣], Qiqing Luo[♣]
[♣]School of Computer Science and Engineering, Southeast University, China
[♠]Key Laboratory of New Generation Artificial Intelligence Technology and Its
Interdisciplinary Applications (Southeast University), Ministry of Education, China
{liguozheng, kewenjun, pwang, xuzijie, keji, jiajliu, ziyus1999, qqluo}@seu.edu.cn

Abstract

The in-context learning (ICL) for relational triple extraction (RTE) has achieved promising performance, but still encounters two key challenges: (1) how to design effective prompts and (2) how to select proper demonstrations. Existing methods, however, fail to address these challenges appropriately. On the one hand, they usually recast RTE task to text-to-text prompting formats, which is unnatural and results in a mismatch between the output format at the pre-training time and the inference time for large language models (LLMs). On the other hand, they only utilize surface natural language features and lack consideration of triple semantics in sample selection. These issues are blocking improved performance in ICL for RTE, thus we aim to tackle prompt designing and sample selection challenges simultaneously. To this end, we devise a tabular prompting for RTE (TableIE) which frames RTE task into a table generation task to incorporate explicit structured information into ICL, facilitating conversion of outputs to RTE structures. Then we propose instructive in-context learning (I^{2}CL) which only selects and annotates a few samples considering internal triple semantics in massive unlabeled samples. Specifically, we first adopt off-the-shelf LLMs to perform schema-agnostic pre-extraction of triples in unlabeled samples using TableIE. Then we propose a novel triple-level similarity metric considering triple semantics between these samples and train a sample retrieval model based on calculated similarities in pre-extracted unlabeled data. We also devise three different sample annotation strategies for various scenarios. Finally, the annotated samples are considered as few-shot demonstrations in ICL for RTE. Experimental results on two RTE benchmarks show that I^{2}CL with TableIE achieves state-of-the-art performance compared to other methods under various few-shot RTE settings.

Keywords: Large language models, relation extraction, in-context learning

1. Introduction

Relational triple extraction (RTE) aims to identify structured information from raw text, involving diverse output structures like named entity recognition (NER) (Zhang et al., 2021; Liu et al., 2024b; Wu et al., 2024) and relation extraction (RE) (Li et al., 2022, 2023b; Wang et al., 2023a,b). Previous studies (Paolini et al., 2021; Lu et al., 2022) propose unified frameworks to tackle the RTE task by converting the structured relational triples into unstructured strings and then utilizing text generation models (Lewis et al., 2020; Raffel et al., 2020). Recent studies (Wei et al., 2022; Wang et al., 2023c; Shang et al., 2024) on large-scale pre-trained language models (LLMs), such as GPT-3 (Brown et al., 2020), demonstrate that LLMs perform well in various natural language processing (NLP) tasks without any fine-tuning but only with a few annotated samples as prompts, which is called *in-context learning* (ICL) (shown in Figure 1 (a)). However, current ICL for RTE still encounters two challenges. On the one hand, recasting RTE task to solely text-to-text prompting format (see Figure 2 TextIE (Ma et al., 2023)) like other NLP tasks is unnatural and resulting in a mismatch between the output format at the pre-training time and the inference time for LLMs (Li et al., 2023c). Therefore, extracting structured data containing multiple dependent elements using text-to-text prompting format makes RTE particularly challenging in ICL. Thus one important question is *how to design proper prompting formats suitable in ICL for RTE?* On the other hand, prior work (Zhao et al., 2021; Liu et al., 2021) has found that the performance of ICL is sensitive to the selected samples. Moreover, due to limited context length of LLMs, only a few annotated samples can be presented in prompts. Therefore, another essential research question is *how to select a few high quality annotated samples as few-shot demonstrations in ICL for RTE?*

Most studies (Lu et al., 2022; Jimenez Gutierrez et al., 2022; Ma et al., 2023; Wan et al., 2023; Liu et al., 2024a) overlook the first challenge and directly transform the RTE task into text-to-text generation formats like TextIE, producing fragile predicted outputs that require complex decoding strategies for post-processing them into valid structures. CodeIE (Li et al., 2023c) (see Figure 2) argues the abundant structured code information en-

LREC-COLING 2024, pages 17131–17143
20-25 May, 2024. © 2024 ELRA Language Resource Association: CC BY-NC 4.0

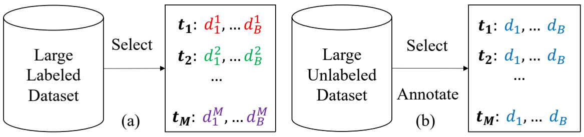


Figure 1: Illustration of the instance-wise retrieving ICL (a) and our I^{2}CL framework (b). For different test samples (e.g., t_1 and t_2), the former paradigm retrieves B different samples from the large labeled dataset as corresponding demonstrations d, while I^{2}CL only selects and annotates a few samples with annotation budget B for all M test samples.

coded in the pre-trained LLMs can benefit RTE task and achieves superior results compared to TextIE. Despite delivering the significant improvement using code LLMs like Codex (Chen et al., 2021), its advantage on natural language LLMs like GPT-3 is slightly inferior. Besides, CodeIE inevitably generates more tokens compared to TextIE as shown in Figure 2, resulting in higher costs and lower efficiency. To this end, we devise a tabular prompting for RTE named TableIE that generates organized and concise outputs with lower costs and higher efficiency, incorporating explicit structured table information into ICL. Specifically, a table header "|*step*|*predicate*|*subject type*|*subject*|*object type*|*object*" is provided as part of the prompt and the LLMs can automatically generate a table, where "|" is the recognizable delimiter of tables in OpenAI [1] models. Specially, TableIE is suitable for both zero and few-shot prompting while TextIE and CodeIE can only be applied to few-shot setting as they cannot guarantee the structural integrity under zero-shot prompting in ICL for RTE.

For the second challenge, we argue that two issues need to be considered. First, despite retrieving the annotated samples as demonstrations in large labeled dataset (see Figure 1 (a)), LLMs ICL still significantly underperforms fine-tuned moderate-size models (Ma et al., 2023). Therefore, considering ICL in zero or low-resource rather than high-resource scenarios is more suitable and promising (Su et al., 2023). In this work, we formulate an instructive in-context learning (I^{2}CL) framework for RTE (shown in Figure 1 (b)): select and annotate a few high-quality samples from the unlabeled data based on the accessible test data, so as to obtain better few-shot prompting results. Second, raw-input-based sample selection method is one widely applied solution in ICL which involves embedding raw inputs of samples using an off-the-shelf embedding model and then selecting the most similar samples (Rubin et al., 2021). Nevertheless, this method is prone to being biased by surface natural language features (syntax, lexicon, semantic, etc.) that may not hold distinct effectiveness for intended tasks (An et al., 2023). In RTE, raw-input-based selection just finds out samples with similar whole sentence semantics, while the better in-context samples should contain the similar or exactly same entity types and relation types (Wan et al., 2023).

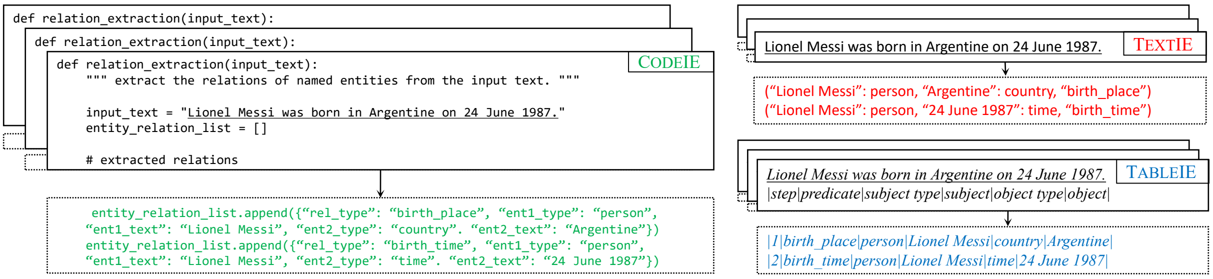
To measure the similarities between samples, features of both entities and relations are required, besides the similarity metrics. First, the unlabeled samples contribute limited unsupervised features. Thus we incorporate the strong capability of LLMs on zero-shot prompting (Kojima et al., 2022) to perform schema-agnostic pre-extraction of entities and relations in unlabeled samples using TableIE. The extracted triples is not consistent with annotation schema, but still objectively represent the semantics of the triples contained in samples. Second, we propose a novel triple-level similarity metric considering the importance of entity and relation types, describing the similarities between pre-extracted unlabeled samples via Pompeiu–Hausdorff distance (Schutze et al., 2012) because a sample may contain multiple triples. After the above two steps, we obtain the similarities between all unlabeled samples pairwise and then fine-tune a Sentence-BERT (Reimers and Gurevych, 2019) as sample retriever on the whole similarities calculated unlabeled data, in which the model pays attention to the internal triple semantic differences between samples. During testing, we adopt this model to calculate the similarities between test and unlabeled data. Then we propose three different sample selection strategies including top-k-based, balance-based and coverage-based strategy, where different strategies are suitable for different test data distributions. Finally, we annotate the selected unlabeled samples as demonstrations. In summary, the contributions of our work are three-fold:

- We propose a tabular prompting TableIE, incorporating explicit structured table information into ICL that achieves superior performance with lower costs and higher efficiency compared to TextIE and CodeIE. Specially, TableIE can be utilized for both zero and few-shot prompting scenarios in LLMs.

- We propose an instructive in-context learning (I^{2}CL) framework, which selects and annotates a few high-quality samples for better prompting results. We propose a novel triple-level similarity metric for instructive sample retrieval and devise three different sample selection strategies suitable for different data distributions.

- Experimental results demonstrate that TableIE performs better than TextIE and CodeIE, indicating the advantage of representing struc-

[1] https://openai.com



```
def relation_extraction(input_text):
def relation_extraction(input_text):
def relation_extraction(input_text):
    """ extract the relations of named entities from the input text. """

    input_text = "Lionel Messi was born in Argentine on 24 June 1987."
    entity_relation_list = []

    # extracted relations
```

```
 entity_relation_list.append({"rel_type": "birth_place", "ent1_type": "person",
"ent1_text": "Lionel Messi", "ent2_type": "country". "ent2_text": "Argentine"})
entity_relation_list.append({"rel_type": "birth_time", "ent1_type": "person",
"ent1_text": "Lionel Messi", "ent2_type": "time". "ent2_text": "24 June 1987"})
```

Lionel Messi was born in Argentine on 24 June 1987.

("Lionel Messi": person, "Argentine": country, "birth_place")
("Lionel Messi": person, "24 June 1987": time, "birth_time")

Lionel Messi was born in Argentine on 24 June 1987.
|step|predicate|subject type|subject|object type|object|

|1|birth_place|person|Lionel Messi|country|Argentine|
|2|birth_time|person|Lionel Messi|time|24 June 1987|

Figure 2: Formats of three prompting. The test sample is marked with underline. The outputs of LLMs are highlighted in colors.

tured targets with table. With the same annotation budget, I^{2}CL consistently improves the naive baselines by a notable margin.

2. Related Work

Zero and Few-shot Prompting The pre-trained LLMs such as GPT-3 (Brown et al., 2020) have demonstrated impressive zero and few-shot learning capabilities across various NLP tasks (Zhao et al., 2021; Liu et al., 2021; Wei et al., 2022; Wang et al., 2023c; Kojima et al., 2022; An et al., 2023). Recent studies (Jimenez Gutierrez et al., 2022; Ma et al., 2023; Wan et al., 2023) on ICL mainly focuses on exploring NER and RE tasks separately, however, there has been relatively little research into potential of LLMs for RTE task (Wei et al., 2023; Li et al., 2023c). ChatIE (Wei et al., 2023) transforms the zero-shot RTE task into a multi-turn question answering problem with a two-stage framework, even surpasses some full shot models on several datasets. CodeIE (Li et al., 2023c) highlights the beneficial of abundant structured code information encoded in LLMs like Codex (Chen et al., 2021) in information extraction (IE) tasks, delivering superior performance compared to common text-to-text few-shot prompting. To the best of our knowledge, we are the first to devise tabular prompting for RTE.

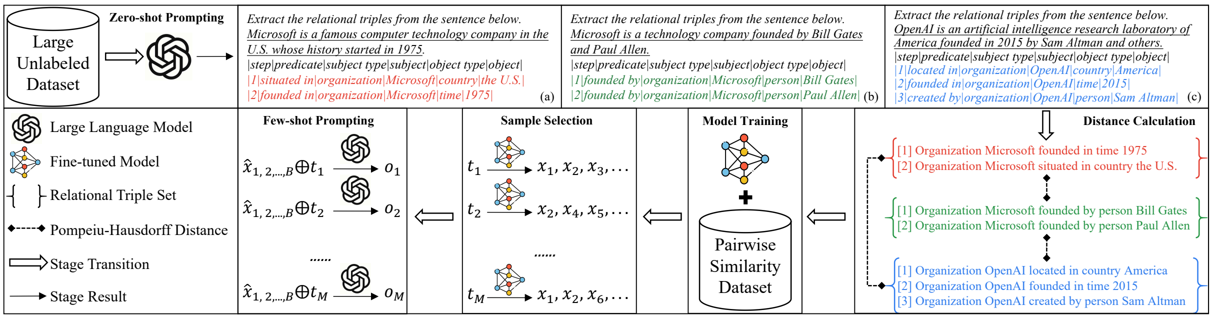
Sample Selection and Annotation Prior studies (Zhao et al., 2021; Liu et al., 2021; An et al., 2023; Liu et al., 2023) have found that the performance of ICL is sensitive to the selected samples as few-shot demonstrations. In addition, retrieving annotated samples on the large labeled dataset still significantly underperforms fine-tuned models in NER and RE tasks (Jimenez Gutierrez et al., 2022; Ma et al., 2023; Wan et al., 2023). Different from the above ICL paradigm, selective annotation (Su et al., 2023) chooses a pool of samples to annotate from unlabeled data in advance, followed by sample retrieval that retrieves task samples from the annotated pool at test time. However, selective annotation computes the similarities between unlabeled samples using Sentence-BERT (Reimers and Gurevych, 2019), where how to select and annotate samples on specific NLP tasks such as RTE is not studied. Moreover, selective annotation maintains a moderate-size sample pool for subsequent sample retrieval during test phase, while I^{2}CL selects a specific number of samples that are most worth annotating based on the test data. Therefore, I^{2}CL is more suitable for RTE and comes at a lower annotation cost compared to selective annotation.

Active Learning Our method for ICL shares the same goal of reducing the annotation cost compared to active learning. For example, iterative parameter updates (Wang et al., 2016; Kasai et al., 2019) based active learning methods are computationally expensive for LLMs used in ICL compared to our method. Recently, the effectiveness of active learning has been questioned when LLMs are fine-tuned for various tasks (Karamcheti et al., 2021). Our method reduces the annotation cost of ICL, departing from the recent observations on fine-tuning with active learning. The limitations of supervised state-of-the-art models for relation extraction in data-scarce and domain-specific scenarios are discussed in Mallart et al. (2022), which is similar with our motivation.

3. Methodology

3.1. Problem Statements

In this work, we assume no large-scale labeled data is available, and we require to select a few samples from existing unlabeled data and annotate them for effective inference. Formally, the objective of I^{2}CL is to select a small subset $\mathcal{S} \subset \mathcal{X}$ from the set of unlabeled samples $\mathcal{X} = \{x_i\}_{i=1}^{N}$ for annotating contained entities and relations based on the test samples $\mathcal{T} = \{t_i\}_{i=1}^{M}$, satisfy the annotation budget $B = |\mathcal{S}|$. N and M represent the total number of samples in unlabeled data and test data, respectively. After the selection and annotation, we treat these samples as the few-shot demonstra-


Zero-shot Prompting
Large Unlabeled Dataset
(a)
(b)
(c)
Large Language Model
Fine-tuned Model
Relational Triple Set
Pompeiu-Hausdorff Distance
Stage Transition
Stage Result
Few-shot Prompting
$\hat{x}_{1,2,\ldots,B} \oplus t_1 \rightarrow o_1$
$\hat{x}_{1,2,\ldots,B} \oplus t_2 \rightarrow o_2$
......
$\hat{x}_{1,2,\ldots,B} \oplus t_M \rightarrow o_M$
Sample Selection
$t_1 \rightarrow x_1, x_2, x_3, \ldots$
$t_2 \rightarrow x_2, x_4, x_5, \ldots$
......
$t_M \rightarrow x_1, x_2, x_6, \ldots$
Model Training
Pairwise Similarity Dataset
Distance Calculation

Extract the relational triples from the sentence below.
Microsoft is a famous computer technology company in the U.S. whose history started in 1975.
|step|predicate|subject type|subject|object type|object|
|1|situated in|organization|Microsoft|country|the U.S.|
|2|founded in|organization|Microsoft|time|1975|

Extract the relational triples from the sentence below.
Microsoft is a technology company founded by Bill Gates and Paul Allen.
|step|predicate|subject type|subject|object type|object|
|1|founded by|organization|Microsoft|person|Bill Gates|
|2|founded by|organization|Microsoft|person|Paul Allen|

Extract the relational triples from the sentence below.
OpenAI is an artificial intelligence research laboratory of America founded in 2015 by Sam Altman and others.
|step|predicate|subject type|subject|object type|object|
|1|located in|organization|OpenAI|country|America|
|2|founded in|organization|OpenAI|time|2015|
|3|created by|organization|OpenAI|person|Sam Altman|

[1] Organization Microsoft founded in time 1975
[2] Organization Microsoft situated in country the U.S.

[1] Organization Microsoft founded by person Bill Gates
[2] Organization Microsoft founded by person Paul Allen

[1] Organization OpenAI located in country America
[2] Organization OpenAI founded in time 2015
[3] Organization OpenAI created by person Sam Altman

Figure 3: Illustration of the I^{2}CL framework. We aim to measure the similarities of the triple sets contained in two samples and select the most representative samples to annotate based on the whole test data. For instance, sample (b) is very similar to (a) on the surface natural language features while (c) is more similar to (a) in the triple-level semantic features. Moreover, annotate one sample (c) is better than two samples (a) and (b) because (c) contains all the similar triple patterns in (a) and (b).

tions in ICL for RTE. We should note that this kind of setting (obtaining the test set in advance to select instances) is practical in real-world scenarios.

3.2. Framework Overview

The framework of I^{2}CL is illustrated in Figure 3, which consists of five stages: (1) zero-shot prompting, (2) distance calculation, (3) model training, (4) sample selection and (5) few-shot prompting. First, we adopt an off-the-shelf LLM to perform schema-agnostic pre-extraction of entities and relations in the unlabeled samples using TableIE. Second, we calculate the similarities between these unlabeled samples based on the pre-extracted triples via average Pompeiu-Hausdorff distance. Third, we train an efficient sample retrieval model in ICL for RTE. During testing, we calculate the similarities between test samples and unlabeled samples. Then we select and annotate B samples by three different strategies and treat them as the few-shot demonstrations. Below we present each stage in detail.

3.2.1. Zero-shot Prompting

To obtain the impartial triple-level semantic features inside large unlabeled data, we utilize the capability of LLMs on zero-shot learning to extract all the relational triples including predicates, subject types, subjects, object types and objects. Specifically, we have the following TableIE in zero-shot prompting format:

Extract the relational triples from the sentence below.
<Sentence>
|step|predicate|subject type|subject|object type|object|

Typically, any valid zero-shot prompting format is compatible with this stage. Unfortunately, existing prompting such as TextIE and CodeIE can only be applied to few-shot prompting, where they are unable to provide precise instructed signals similar to table header in TableIE for RTE.

3.2.2. Distance Calculation

After pre-extracting, we need to measure the similarities between the unlabeled samples on pre-extracted triple sets level. Specifically, we first transform each triple containing predicate p, subject type s_t, subject s, object type o_t and object o as the natural language form z:

$$z = s_t \oplus s \oplus p \oplus o_t \oplus o \quad (1)$$

where $\oplus$ indicates simple concatenation of strings. The intuition behind our method is that the types of entities and relations are important manifestations of triplet-level semantics, rather than just considering the span of entities. Formally, we define the distance between the triples z_i and z_j as:

$$d(z_i, z_j) = ||\mathrm{Encoder}(z_i) - \mathrm{Encoder}(z_j)||_2 \quad (2)$$

where $\mathrm{Encoder}(\cdot)$ denotes the encoder-only pre-trained language model such as Sentence-BERT (Reimers and Gurevych, 2019) and $||\cdot||_2$ denotes the Euclidean distance. Suppose we have two samples with triple sets $\mathcal{Z}_i$ and $\mathcal{Z}_j$, we aim to obtain the Pompeiu-Hausdorff distance between them to measure the similarities by considering all the relational triples simultaneously. As the standard Pompeiu-Hausdorff distance is highly sensitive to outliers, we use the average Pompeiu-Hausdorff distance (Schutze et al., 2012):

$$\begin{aligned} D(\mathcal{Z}_i, \mathcal{Z}_j) = & \frac{1}{|\mathcal{Z}_i|} \sum_{z_i \in \mathcal{Z}_i} \min_{z_j \in \mathcal{Z}_j} d(z_i, z_j) + \\ & \frac{1}{|\mathcal{Z}_j|} \sum_{z_j \in \mathcal{Z}_j} \min_{z_i \in \mathcal{Z}_i} d(z_i, z_j) \end{aligned} \quad (3)$$

where the triple sets distance $D(\mathcal{Z}_i, \mathcal{Z}_j)$ serves as the distance of two unlabeled samples in RTE task.

3.2.3. Model Training

After obtaining the similarities between all unlabeled samples pairwise, we aim to train an efficient sample retriever for test samples in ICL for RTE. Apparently, we are able to extract the entities and relations from each test sample via LLMs as before, and measure the similarities between unlabeled samples and test samples for selective annotations. However, it is very inconvenient and costly to extract triples from each test sample in advance when new test data arrives, especially with a large number of test samples. To this end, we train an encoder-based model $f_\theta(\cdot)$ which considers solely surface natural language features as inputs but pays more attention to learn the internal relational triple patterns between different samples. Specifically, we fine-tune a Sentence-BERT (Reimers and Gurevych, 2019) on the similarities calculated unlabeled data which aims to approximately obtain the distance between samples x_i and x_j via:

$$\mathcal{L} = \sum_{(x_i,x_j)\in\mathcal{X}} (\mathcal{D}(\mathcal{Z}_i,\mathcal{Z}_j) - ||f_\theta(x_i) - f_\theta(x_j)||_2)^2 \quad (4)$$

where $f_\theta(x)$ denotes the sentence representation of unlabeled sample x. We minimize the differences between the values of Pompeiu-Hausdorff distances and sentence representation distances. During testing, we adopt the fine-tuned model to calculate the distances between each test sample and unlabeled sample so as to obtain the pairwise distance set $\mathcal{P} = \{p_{(i,j)}\}_{(i,j)=(1,1)}^{N\times M}$ between unlabeled samples $\mathcal{X}$ and test samples $\mathcal{T}$:

$$\mathcal{P} = \{p_{(i,j)} \,|\, p_{(i,j)} = d(x_i,t_j), x_i \in \mathcal{X}, t_j \in \mathcal{T}\} \quad (5)$$

where $d(x_i,t_j) = ||f_\theta(x_i) - f_\theta(t_j)||_2$.

3.2.4. Sample Selection

In sample selection, we select and annotate samples from unlabeled data with annotation budget B. We propose three different sample selection strategies based on the pairwise distance set $\mathcal{P}$.

- **Top-k based strategy** We select the most similar u unlabeled samples for each test sample t, resulting in a total of $M \times u$ unlabeled samples. Then we count the frequency of each unlabeled sample occurrence and sort it in descending order of frequency. Finally, we directly select and annotate the first k unlabeled samples for ICL. In this case, we have $k = B$.

- **Balance-based strategy** The top-k method is simple but may cause imbalanced labeling, especially when B is relatively small, which may result in a lack of samples of some specific entity or relation types. If the test data contains R relation types, we select and annotate B samples with R-way $\lfloor B/R \rfloor$-shot style in the sorted unlabeled samples. Note this strategy possibly increases the annotation cost, especially under the adverse effects of imbalanced unlabeled data distribution in relation types.

- **Coverage-based strategy** The above two strategies cannot guarantee that the selected unlabeled samples are similar to all the test samples. We thus propose a coverage-based sample selection strategy that ensure the participation of all test samples in selective annotation process, which is described in Algorithm 1. We first select the nearest $\lceil M/B \rceil$ test samples for each unlabeled sample x based on pairwise distance set $\mathcal{P}$ (line 1-8). Then the unlabeled sample x_b with the minimum sum of pairwise distances is selected to annotate (line 9-10). Subsequently, we discard the unlabeled sample x_b and its nearest $\lceil M/B \rceil$ test samples in pairwise distance set $\mathcal{P}$ (line 11-12). We iterate through the above process until all the test samples are discarded.

Algorithm 1 Coverage-based strategy for sample selection

```
Input: The unlabeled samples $\mathcal{X} = \{x_i\}_{i=1}^{N}$, test samples $\mathcal{T} = \{t_i\}_{i=1}^{M}$ and pairwise distance set $\mathcal{P} = \{p_{(i,j)}\}_{(i,j)=(1,1)}^{N\times M}$.
Output: The selective annotation subset $\mathcal{S} \subset \mathcal{X}$ with annotation budget $B = |\mathcal{S}|$.
 1: $\mathcal{S} = \varnothing$
 2: for $s \leftarrow 1$ to $B$ do
 3:     if $\forall t_j \in \mathcal{T}$, $p_{(*,j)} \notin \mathcal{P}$ then
 4:         return $\mathcal{S}$
 5:     else
 6:         for $x_i \in \mathcal{X}$, $p_{(i,*)} \in \mathcal{P}$ do
 7:             $\mathcal{P}_i = \{p_{(i,*)} \,|\, p_{(i,*)} \in \min(\mathcal{P}, \lceil M/B \rceil)\}$
 8:         end for
 9:         $x_b = \arg\min_{x_i \in \mathcal{X}} \sum_{*}^{\lceil M/B \rceil} p_{(i,*)}$ where $p_{(i,*)} \in \mathcal{P}_i$
10:         $\mathcal{S} \leftarrow \mathcal{S} \cup x_b$
11:         Discard $p_{(i,*)}$ in $\mathcal{P}$ where $x_i = x_b$
12:         Discard $p_{(*,j)}$ in $\mathcal{P}$ where $p_{(b,j)} \in \mathcal{P}_b$
13:     end if
14: end for
```

3.2.5. Few-shot Prompting

After selecting and annotating $\mathcal{S} = \{(x_i, y_i)\}_{i=1}^{B}$, we adopt standard few-shot prompting for ICL in RTE. We convert the samples in $\mathcal{S}$ to corresponding table-style pairs $\{(\hat{x}_i, \hat{y}_i)\}_{i=1}^{B}$. Then we concatenate them as a string to compose the in-context demonstrations $\hat{x}_{1,\ldots B} = I \oplus \hat{x}_1 \oplus \hat{y}_1 \ldots \hat{x}_B \oplus \hat{y}_B$, where I denotes the prompt instructions as *Extract the relational triples from the sentences below.* Note that we arrange these demonstrations in ascending order of similarities to the test samples $\mathcal{T}$.

Given a specific test sample t_i, after feeding the constructed input into the LLMs, we are expected to get an output that is formatted as the same as $y_1, ..., y_B$ which retains the integral table structural:

$$o_i = \mathrm{LLMs}\left(\hat{x}_{1,...,B} \oplus t_i\right) \tag{6}$$

where $\hat{x}_{1,...,B} \oplus t_i$ is the prompt for each test sample $t_i \in \mathcal{T}$ and o_i is the extracted relational triples.

4. Experiments

4.1. Experimental Design

4.1.1. Datasets

We evaluate our proposed TableIE and I^{2}CL in RTE task with benchmarks CoNLL04 (Roth and Yih, 2004) and NYT (Riedel et al., 2010) where Table 1 shows the dataset statistics. We follow Lu et al. (2022) and Li et al. (2023c) to preprocess all these datasets. Then we remove all annotated labels from the training set and treat them as unlabeled data. We select and annotate the fake unlabeled training set based on its test set in each dataset.

Table 1: Statistics of the datasets. # Ents and # Rels denote the number of entity types and relation types. # Train, # Valid and # Test denote the sample number in each split.

Dataset	# Ents	# Rels	# Train	# Valid	# Test
CoNLL04	4	5	922	231	288
NYT	3	24	56,196	5,000	5,000

4.1.2. Settings

For prompting formats of TextIE (Ma et al., 2023), CodeIE (Li et al., 2023c) and our TableIE, we use the same backbone of LLMs including the variant of GPT-3 (Brown et al., 2020) "`text-davinci-003`", ChatGPT [2] "`gpt-3.5-turbo`" and GPT-4 (OpenAI, 2023) "`gpt-4`". Note that CodeIE performs well on Codex (Chen et al., 2021), but unfortunately Codex is deprecated by OpenAI. Typically, we get model predictions by querying OpenAI API in few-shot prompting manner. The hyper-parameters of LLMs with three promptings are keep consistent.

In I^{2}CL, we train the sample retriever $f_\theta(\cdot)$ for 5 epochs with batch size 16 and learning rate 2e-5 on the pre-extracted unlabeled data using the AdamW (Loshchilov and Hutter, 2019) optimizer. And 10% of samples in unlabeled data are regarded as the validation data to select the best model $f_\theta(\cdot)$. For the annotation budget B, we experiment with 5, 15 and 25 on CoNLL04, respectively. Due to the context length limit, we are unable to experiment above LLMs with a large budget. Thus we adopt the "`gpt-3.5-turbo-16k`" model to support larger budget 24, 48 and 72 on NYT, respectively. With the same budget, we experiment with our proposed three sample selection strategies: `top-k` (u = 5), `balance` and `coverage` with proposed TableIE. For TextIE and CodeIE, we randomly sampling k training samples for each relation type to construct a k-shot demonstration set. We also test the TableIE with randomly selected samples and replace the sample retriever $f_\theta(\cdot)$ with original Sentence-BERT [3] (Reimers and Gurevych, 2019) to verify the effectiveness of I^{2}CL framework.

Following previous work (Lu et al., 2022; Li et al., 2023c), we use the relation strict F1 as the evaluation metrics for the results of RTE. Specifically, a relational triple prediction is correct only if the relation type is correct and the corresponding offsets and types of its entities are correct. Due to the high variance of few-shot prompting with the random selection method, we conduct three runs with different random seeds for each experiment and report the mean values.

4.2. Main Results

As shown in Table 2, TableIE outperforms other prompting formats and equipping I^{2}CL with TableIE in various LLMs (GPT-3, ChatGPT and GPT-4) consistently achieve superior performance over typical ICL under few-shot settings on CoNLL04, demonstrating the effectiveness of our proposed TableIE prompting and I^{2}CL framework. On the one hand, TableIE outperforms TextIE under different experimental settings and achieves competitive or superior performance compared to CodeIE, which highlights the importance and beneficial of incorporating explicit structure information into RTE task. On the other hand, I^{2}CL with `balance` and `coverage` delivers significantly improvement compared to TableIE with random selection, which indicates that appropriate sample selection strategies provide instructive suggestions for annotating representative samples in ICL for RTE. Concretely, `balance` and `coverage` achieve similar performance while the latter is suitable for larger annotation budgets. However, `top-k` fails to achieve promising results on CoNLL04 especially when the annotation budget is relatively small. We provide a concrete example to analyze this phenomenon. Intuitively, `top-k` is highly affected by the distribution of test data relation types, where 47 relational triples in CoNLL04 test data belongs to relation *Kill* and 105 belongs to *OrgBased_In*. Empirically, we discover that there are no relational triples belonging to *Kill* with annotated budget 5. With the increase of anno-

[2] https://openai.com/blog/chatgpt

[3] `all-mpnet-base-v2`

Table 2: Experimental results on CoNLL04 benchmark. Best results with different budgets and models are in **bold**.

Model	Method	CoNLL04		
		B=5	B=15	B=25
GPT-3	TextIE	19.85	32.83	40.35
	CodeIE	36.23	41.29	49.98
	TableIE	**37.36**	**42.90**	**50.77**
	+ I^{2}CL$_{\texttt{top-k}}$	25.97 ↓	39.49 ↓	47.71 ↓
	+ I^{2}CL$_{\texttt{balance}}$	**39.30** ↑	**46.77** ↑	54.97 ↑
	+ I^{2}CL$_{\texttt{coverage}}$	36.21 ↑	45.41 ↑	**55.36** ↑
ChatGPT	TextIE	23.55	37.53	42.95
	CodeIE	36.83	42.87	49.78
	TableIE	**37.29**	**43.15**	**50.31**
	+ I^{2}CL$_{\texttt{top-k}}$	27.70 ↓	43.56 ↑	49.72 ↓
	+ I^{2}CL$_{\texttt{balance}}$	**40.18** ↑	47.24 ↑	55.33 ↑
	+ I^{2}CL$_{\texttt{coverage}}$	36.35 ↑	**47.38** ↑	**56.45** ↑
GPT-4	TextIE	24.92	38.35	46.53
	CodeIE	36.77	43.25	50.19
	TableIE	**37.80**	**43.81**	**51.38**
	+ I^{2}CL$_{\texttt{top-k}}$	28.41 ↓	44.11 ↑	50.44 ↓
	+ I^{2}CL$_{\texttt{balance}}$	**40.51** ↑	**48.93** ↑	57.54 ↑
	+ I^{2}CL$_{\texttt{coverage}}$	37.63 ↑	48.70 ↑	**58.12** ↑

tated budget, the adverse effects brought by `top-k` have been significantly alleviated, but it is still inferior to random selection on CoNLL04. Moreover, when using the same kind of prompting and comparing the used LLMs, `gpt-4` demonstrates stronger extraction capability than the other two LLMs `text-davinci-003` and `gpt-3.5-turbo`, but the performance gap between the three is not notable, especially when experiment with less annotated data and random selection strategy. Generally, the sample selection strategies improve final ICL performance by 4.59% on `text-davinci-003` and 6.74% on `gpt-4` compared to `random`. Without sample selection, only 0.44%, 0.91% and 0.61% improvement gained compared to `text-davinci-003` for `gpt-4` with B = 5, 15 and 25, respectively. Thus we argue that I^{2}CL provide instructive sample annotation and treat these representative samples as few-shot demonstrations stimulates LLMs stronger capability of understanding relational triples.

We further compare these approaches under different annotation budget on NYT. As shown in Table 3, we can see that the discovered phenomenons on CoNLL04 still hold except for the differences in performance among the three sample selection strategies. We notice that `top-k` can achieve much better performance compared to its counterparts especially with larger annotation budget, while `balance` only deliver similar results compared to random selection. Here we provide the empirical analysis. Specifically, NYT is an extremely imbalanced dataset where the imbalance of samples belonging to different relations makes it difficult to select samples with appropriate proportions of different relations in ICL. Note that */location/location/contains* is the most frequently appearing relation in NYT test data, where total 3,827 of 8,110 relational triples belonging to this relation. In contrast, some relations such as */people/person/ethnicity* and */people/ethnicity/geographic_distribution* correspond to very few samples (i.e., only one sample) in whole test data. When selecting demonstrations in ICL for such extremely imbalanced test data, it is unreasonable to sample k samples for each relation to construct a k-shot demonstration set. Typically, we cannot know in advance the proportion of each relation in the test set, and using `balance` may not necessarily achieve good results. Therefore, we should choose the best strategy based on the distribution of test data, where `coverage` is stabler and better than other two strategies and consistently achieves satisfactory results with insensitivity to test data distribution.

Table 3: Experimental results on NYT benchmark. Best results with different budgets and models are in **bold**. The results are all based on the `gpt-3.5-turbo-16k`.

Model	Method	NYT		
		B=24	B=48	B=72
ChatGPT	TextIE	18.85	18.88	19.44
	CodeIE	28.23	28.75	29.78
	TableIE	**29.31**	**29.84**	**30.45**
	+ I^{2}CL$_{\texttt{top-k}}$	**32.28** ↑	**34.92** ↑	**35.66** ↑
	+ I^{2}CL$_{\texttt{balance}}$	29.23 ↓	30.24 ↑	30.42 ↓
	+ I^{2}CL$_{\texttt{coverage}}$	31.44 ↑	32.21 ↑	32.89 ↑

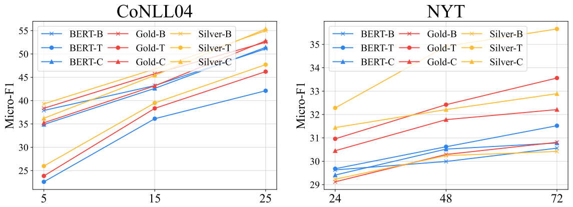


Figure 4: Performance of different retrieval models on two benchmarks. We use `text-davinci-003` on CoNLL04 and `gpt-3.5-turbo-16k` on NYT. BERT denotes the original Sentence-BERT without fine-tuning. Gold denotes the fine-tuned model on annotated training data, while Silver denotes the fine-tuned model on training data with pre-extraction results. And B, T, C represent `balance`, `top-k`, and `coverage` strategies, respectively.

Table 4: Model generalization results on CoNLL04 and NYT. Best results are in **bold**. Results that have improved compared to the original ones are marked with underline.

Method	CoNLL04			NYT		
	B=5	B=15	B=25	B=24	B=48	B=72
TableIE	37.36	42.90	50.77	29.31	29.84	30.45
+ I^{2}CL$_{\texttt{top-k}}$	28.97	40.71	49.31	31.36	**32.62**	**33.79**
+ I^{2}CL$_{\texttt{balance}}$	**37.88**	**44.37**	53.12	29.10	30.14	30.20
+ I^{2}CL$_{\texttt{coverage}}$	35.78	43.40	**53.26**	**31.43**	31.56	33.21

Table 5: Costs and efficiency results. # Total denotes the total number of characters in LLMs outputs, and # Avg., # Min. and # Max. represent the average, minimum and maximum values of character lengths of all samples, respectively.

Method	# Total ↓	# Avg. ↓	# Min. ↓	# Max. ↓
TextIE	28,473	98.86	38	527
CodeIE	71,431	248.02	131	2456
TableIE	**24,976**	**86.72**	**30**	**460**

4.3. Different Retrieval Results

To verify the effectiveness of retrieval model in I^{2}CL, we evaluate the impact of different retrieval models on final ICL performance. Besides the proposed retrieval model (Silver), we also adopt the original Sentence-BERT model (BERT) and train a new retrieval model (Gold) based on the original annotated training data. As shown in Figure 4, considering relational triple features and adopting Pompeiu-Hausdorff distance as similarity metric bring better retrieving results. Consistent with the conclusion of the main experiment, `balance` and `coverage` perform better on CoNLL04 while `top-k` is more suitable for NYT. Intuitively, retrieval model based on ground truth would achieve the best results. However, Gold is only slightly better than BERT and worse than Silver on two benchmarks. This counter-intuitive phenomenon is likely due to two reasons. One is that there are many samples with incorrect annotation or incomplete annotation in the training set of datasets, especially the NYT which is completely constructed by distant supervision (Mintz et al., 2009). These inaccurate noise signals can degrade the final performance of retrieval model. Another is that we force the model to learn to retrieve the samples that similar to test samples in triple-level semantics from the perspective of LLMs rather than annotators. Compared to previous approaches, our metric serves as a more accurate proxy for evaluating the utility of a training sample during testing. In contrast, BERT solely retrieves samples with similar sentence semantics but ignore the subtle semantics of entities and relations inside samples, where the improvement of using BERT as retrieval model is modest compared to random selection without any retriever.

4.4. Model Generalization Ability

In this section we explore the retrieval generalization ability. Specifically, we use the retrieval model trained on unlabeled data of NYT on CoNLL04, simulating cases where a retrieval model is directly deployed in a new scenario to test model generalization. We are able to experiment in this manner because the pre-extraction process using zero-shot prompting is schema-agnostic. In other words, we expect the retrieval model to learn the objectively semantic information and can be adapted to new datasets. We conduct experiments with `text-davinci-003` on CoNLL04 and `gpt-3.5-turbo` on NYT both with TableIE, showing results in Table 4. With the exchangeable retrieval model, the improvement from random selection to strategic selection is less than 3%, indicating the data drift issue degrading the model performance to some extent. Under different budget settings, however, I^{2}CL still enjoys obvious advantages regardless of diverse LLMs as backbones. Note that the exchangeable retrieval model also improves the performance of `top-k` on CoNLL04 and `coverage` on NYT, which indicates that our method achieves moderate generalization ability across datasets.

4.5. Costs and Efficiency

Specially, we compare the costs and efficiency of three different prompting formats using `text-davinci-003` on CoNLL04. The experimental results are illustrated in Table 5. Since we are unable to access the specific tokenizer in LLMs, we estimated the characters generated using different methods. Generally, generating more characters (i.e., more tokens) means spending more response time and consuming more computing resources, suffering from very low efficiency. Specifically, the total number of characters generated using CodeIE is around 2.9 × as TableIE, which is not surprising because it requires to generate redundant tokens such as the keys in Python dictionaries. Moreover, the minimum and maximum values of character lengths is 4.37 - 5.34 × as TableIE. Compared with TextIE, our TableIE only generates results with less than 0.88 × average character length, but achieves significantly better performance in ICL for RTE, demonstrating the superior of TableIE.

4.6. Discussions on Settings

Obviously, the proposed method needs to obtain the test set in advance to select instances. We acknowledge that the setting defined in this work is not very common. Generally, we will assume that the test samples are unknown, and then design corresponding algorithms to select ICL samples in advance, as defined in previous work (Su et al., 2023). However, in a lot of practical scenarios, selecting and annotating samples in advance is so ideal and typically brings suboptimal results.

For example, imagine we are developing a system to extract relational triples from news articles. We start with a set of unlabeled news articles. These articles cover various topics, including politics, sports, and entertainment. But we are aware that the model may not perform well on specific and niche topics that are not well-represented in the training data (e.g., the new test samples belong to military topic). As new articles are published, we receive a stream of unlabeled test samples. These test samples may cover emerging events, new personalities, or unique scenarios not present in the initial training data. Instead of pre-annotating a fixed set of demonstrations, we dynamically annotate a few examples from each batch of test samples based on their topics. These annotations serve as in-context demonstrations.

A very important question is: why do we have to choose unchanging samples in advance for ICL when targeting future unknown test samples? We argue that a reasonable solution is selecting then annotating a few good samples as demonstrations for specific test samples in a specific scenario and time. Annotating samples in advance has potential risks: we do not know in advance what entity and relation types test samples will contain, which can lead to pre-annotated samples deviating from test samples in terms of type distribution. In our setting, training a triple-level similarity retriever in a specific domain is valuable, because we can provide personalized demonstrations for every time new test samples in this domain arrive compared to unchanging samples. Specifically, the setting in Su et al. (2023) works well with small distribution deviation between test samples and annotated samples. Our setting expects to determine the distribution of annotated samples based on test samples. In summary, we argue that in a lot of practical scenarios, it is reasonable and sometimes necessary to annotate a few high quality samples with less human labor for potentially massive test samples to be predicted, considering the performance requirements in real-world scenarios.

4.7. Beyond GPT-Series Models

Besides the OpenAI GPT-Series LLMs, we also consider other LLMs such as LLaMA (Touvron et al., 2023), T5 (Raffel et al., 2020), OPT (Zhang et al., 2022) etc. However, we empirically find these open-source LLMs basically cannot perform zero and few-shot prompting in RTE similar with the findings in Li et al. (2023a) perhaps due to the LLMs are not large enough to perform ICL. For example, we tried to experiment LLaMA-7B on RTE task with several demonstrations via few-shot prompting paradigm. However, we observe that LLaMA cannot understand the instructions and is unable to recover the expected triples based on demonstrations, and LLaMA just starts to completely repeat the input sentences. Current ICL research in information extraction (IE) only focuses on very large models (Li et al., 2023c; Ma et al., 2023; Li et al., 2023a) because smaller foundation models are unable to perform ICL in IE just like simpler NLP tasks such as sentiment analysis.

4.8. Why TableIE Works

Foremost, TEXIE and CODEIE cannot guarantee the structural integrity under zero-shot prompting because there is no explicit prompt (i.e., table header) to guide them in generating valid triples without few-shot demonstrations. Since prompting is a brittle process wherein small modifications to the prompt can cause large variations in the model predictions, we are unable to theoretically judge the effectiveness of a prompting other than through empirical results. But we can provide some relevant inspiring works to enhance our conclusion. As the output of RTE is structured, incorporating explicit structured information into ICL tends to benefit the final RTE performance (Li et al., 2023c). Besides, reasoning and extracting answers step by step in such tabular format is effective in other downstream tasks (Ziqi and Lu, 2023). And LLMs such as GPT-3 (Brown et al., 2020) and CodeX (Chen et al., 2021) have the capability of reasoning over tabular structured data, because such models are trained on massive tabular formed data.

5. Conclusion

In this work, we devise a tabular prompting TableIE, framing the RTE task to a table generation task and achieving promising performance in ICL. We propose I^{2}CL, an instructive in-context learning framework for RTE, which is more effective and realistic in zero or low-resource scenarios. I^{2}CL leverages the capability of LLMs on zero-shot prompting and natural language understanding to achieve better few-shot prompting results with less annotation. We also propose a novel triple-level similarity metric for sample retrieval. Besides, three sample selection strategies are proposed to annotate proper samples with a few annotation budget, where we may require to choose the best strategy based on data distribution according to empirical results, encouraging more effective methods in the future research.

6. Bibliographical References

Shengnan An, Bo Zhou, Zeqi Lin, Qiang Fu, Bei Chen, Nanning Zheng, Weizhu Chen, and Jian-Guang Lou. 2023. Skill-based few-shot selection for in-context learning. *arXiv preprint arXiv:2305.14210*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. In *NeurIPS*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Bernal Jimenez Gutierrez, Nikolas McNeal, Clayton Washington, You Chen, Lang Li, Huan Sun, and Yu Su. 2022. Thinking about GPT-3 in-context learning for biomedical IE? think again. In *Findings of EMNLP*.

Siddharth Karamcheti, Ranjay Krishna, Li Fei-Fei, and Christopher D Manning. 2021. Mind your outliers! investigating the negative impact of outliers on active learning for visual question answering. In *ACL*.

Jungo Kasai, Kun Qian, Sairam Gurajada, Yunyao Li, and Lucian Popa. 2019. Low-resource deep entity resolution with transfer and active learning. *arXiv preprint arXiv:1906.08042*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *NeurIPS*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *ACL*.

Guozheng Li, Xu Chen, Peng Wang, Jiafeng Xie, and Qiqing Luo. 2022. Fastre: Towards fast relation extraction with convolutional encoder and improved cascade binary tagging framework. In *IJCAI*.

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

Guozheng Li, Peng Wang, Qiqing Luo, Yanhe Liu, and Wenjun Ke. 2023b. Online noisy continual relation learning. In *AAAI*.

Peng Li, Tianxiang Sun, Qiong Tang, Hang Yan, Yuanbin Wu, Xuanjing Huang, and Xipeng Qiu. 2023c. Codeie: Large code generation models are better few-shot information extractors. In *ACL*.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2021. What makes good in-context examples for gpt-3? *arXiv preprint arXiv:2101.06804*.

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

Jiajun Liu, Peng Wang, Ziyu Shang, and Chenxiao Wu. 2023. Iterde: an iterative knowledge distillation framework for knowledge graph embeddings. In *AAAI*.

Yanhe Liu, Peng Wang, Wenjun Ke, Guozheng Li, Xiye Chen, Jiteng Zhao, and Ziyu Shang. 2024b. Unify named entity recognition scenarios via contrastive real-time updating prototype. In *AAAI*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *ICLR*.

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 *ACL*.

Yubo Ma, Yixin Cao, YongChing Hong, and Aixin Sun. 2023. Large language model is not a good few-shot information extractor, but a good reranker for hard samples! *arXiv preprint arXiv:2303.08559*.

Cyrielle Mallart, Michel Le Nouy, Guillaume Gravier, and Pascale Sébillot. 2022. Confronting active learning for relation extraction to a real-life scenario on french newspaper data. In *InterNLP@ NeurIPS 2022*.

Katerina Margatina, Giorgos Vernikos, Loïc Barrault, and Nikolaos Aletras. 2021. Active learning by acquiring contrastive examples. *arXiv preprint arXiv:2109.03764*.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *ACL*.

OpenAI. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, Rishita Anubhai, Cicero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. 2021. Structured prediction as translation between augmented natural languages. In *ICLR*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*, 21(140):1–67.

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

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

Dan Roth and Wen-tau Yih. 2004. A linear programming formulation for global inference in natural language tasks. In *NAACL*.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2021. Learning to retrieve prompts for in-context learning. In *NAACL*.

Oliver Schutze, Xavier Esquivel, Adriana Lara, and Carlos A Coello Coello. 2012. Using the averaged hausdorff distance as a performance measure in evolutionary multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 16(4):504–522.

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

Hongjin Su, Jungo Kasai, Chen Henry Wu, Weijia Shi, Tianlu Wang, Jiayi Xin, Rui Zhang, Mari Ostendorf, Luke Zettlemoyer, Noah A Smith, et al. 2023. Selective annotation makes language models better few-shot learners. In *ICLR*.

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

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. *arXiv preprint arXiv:2305.02105*.

Keze Wang, Dongyu Zhang, Ya Li, Ruimao Zhang, and Liang Lin. 2016. Cost-effective active learning for deep image classification. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(12):2591–2600.

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

Peng Wang, Jiafeng Xie, Xiye Chen, Guozheng Li, and Wei Li. 2023b. Pascore: a chinese overlapping relation extraction model based on global pointer annotation strategy. In *IJCAI*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, and Denny Zhou. 2023c. Self-consistency improves chain of thought reasoning in language models. In *ICLR*.

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

Xiang Wei, Xingyu Cui, Ning Cheng, Xiaobin Wang, Xin Zhang, Shen Huang, Pengjun Xie, Jinan Xu, Yufeng Chen, Meishan Zhang, et al. 2023. Zero-shot information extraction via chatting with chatgpt. *arXiv preprint arXiv:2302.10205*.

Chenxiao Wu, Wenjun Ke, Peng Wang, Zhizhao Luo, Guozheng Li, and Wanyi Chen. 2024. Consistner: Towards instructive ner demonstrations for llms with the consistency of ontology and context. In *AAAI*.

Dong Zhang, Suzhong Wei, Shoushan Li, Hanqian Wu, Qiaoming Zhu, and Guodong Zhou. 2021. Multi-modal graph fusion for named entity recognition with targeted visual guidance. In *AAAI*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *ICML*.

Jin Ziqi and Wei Lu. 2023. Tab-cot: Zero-shot tabular chain of thought. In *Findings of ACL*.

A. Retrieval with Pompeiu-Hausdorff

We experiment the strategy (we call this strategy "Z+S") of Zero-shot prompting (extracting triples from test samples) + Sentence-BERT (using Sentence-BERT to select unlabeled samples via Pompeiu-Hausdorff similarities), and this strategy could achieve similar or sometimes slightly better results (but typically less than 1%) compared to our fine-tuned retriever, as shown in Table 6 and Table 7. This is reasonable because the aim of fine-tuned retriever is to approximate the selection quality of this strategy but without extra pre-extraction on possibly massive test samples as mentioned in Model Training section. In other words, the fine-tuned retriever is expected to select good demonstrations taking solely surface natural language features as inputs but pays more attention to the internal relational triple patterns during testing.

Table 6: Experimental results on CoNLL04 benchmark. Best results with different budgets and models are in **bold**.

Model	Method	CoNLL04		
		B=5	B=15	B=25
GPT-3	TextIE	19.85	32.83	40.35
	CodeIE	36.23	41.29	49.98
	TableIE	**37.36**	**42.90**	**50.77**
	+ I^{2}CL$_{\texttt{top-k}}$	25.97 ↓	39.49 ↓	47.71 ↓
	+ Z+S$_{\texttt{top-k}}$	25.30 ↓	40.22 ↓	48.64 ↓
	+ I^{2}CL$_{\texttt{balance}}$	39.30 ↑	46.77 ↑	54.97 ↑
	+ Z+S$_{\texttt{balance}}$	**39.38** ↑	**46.96** ↑	55.60 ↑
	+ I^{2}CL$_{\texttt{coverage}}$	36.21 ↑	45.41 ↑	55.36 ↑
	+ Z+S$_{\texttt{coverage}}$	36.42 ↑	45.03 ↑	**55.88** ↑
ChatGPT	TextIE	23.55	37.53	42.95
	CodeIE	36.83	42.87	49.78
	TableIE	**37.29**	**43.15**	**50.31**
	+ I^{2}CL$_{\texttt{top-k}}$	27.70 ↓	43.56 ↑	49.72 ↓
	+ Z+S$_{\texttt{top-k}}$	27.24 ↓	43.50 ↑	50.48 ↑
	+ I^{2}CL$_{\texttt{balance}}$	40.18 ↑	47.24 ↑	55.33 ↑
	+ Z+S$_{\texttt{balance}}$	**40.40** ↑	47.31 ↑	56.10 ↑
	+ I^{2}CL$_{\texttt{coverage}}$	36.35 ↑	47.38 ↑	**56.45** ↑
	+ Z+S$_{\texttt{coverage}}$	36.42 ↑	**47.67** ↑	56.37 ↑
GPT-4	TextIE	24.92	38.35	46.53
	CodeIE	36.77	43.25	50.19
	TableIE	**37.80**	**43.81**	**51.38**
	+ I^{2}CL$_{\texttt{top-k}}$	28.41 ↓	44.11 ↑	50.44 ↓
	+ Z+S$_{\texttt{top-k}}$	28.75 ↓	44.39 ↑	51.04 ↓
	+ I^{2}CL$_{\texttt{balance}}$	40.51 ↑	48.93 ↑	57.54 ↑
	+ Z+S$_{\texttt{balance}}$	**40.62** ↑	49.45 ↑	58.50 ↑
	+ I^{2}CL$_{\texttt{coverage}}$	37.63 ↑	48.70 ↑	58.12 ↑
	+ Z+S$_{\texttt{coverage}}$	37.98 ↑	**49.64** ↑	**58.53** ↑

Table 7: Experimental results on NYT benchmark. Best results with different budgets and models are in **bold**. The results are all based on the `gpt-3.5-turbo-16k`.

Model	Method	NYT		
		B=24	B=48	B=72
ChatGPT	TextIE	18.85	18.88	19.44
	CodeIE	28.23	28.75	29.78
	TableIE	**29.31**	**29.84**	**30.45**
	+ I^{2}CL$_{\texttt{top-k}}$	32.28 ↑	34.92 ↑	35.66 ↑
	+ Z+S$_{\texttt{top-k}}$	**33.12** ↑	**35.64** ↑	**36.27** ↑
	+ I^{2}CL$_{\texttt{balance}}$	29.23 ↓	30.24 ↑	30.42 ↓
	+ Z+S$_{\texttt{balance}}$	29.74 ↑	30.80 ↑	30.67 ↑
	+ I^{2}CL$_{\texttt{coverage}}$	31.44 ↑	32.21 ↑	32.89 ↑
	+ Z+S$_{\texttt{coverage}}$	31.48 ↑	32.94 ↑	33.30 ↑

B. Comparison with Vote-k

We mentioned the vote-k (selective annotation) in our related work but not explicitly discussed it in our experiments because we tend to think this kind of comparison is neither fair or necessary.

First, vote-k and I^{2}CL are applied in different practical settings. The vote-k selects samples to annotate before test time while I^{2}CL selects samples to annotate when new test samples arrive. In other words, vote-k finds a few global representative samples in advance to annotate but may result in suboptimal results for test samples in the future, while I^{2}CL finds local optimal demonstrations for current test samples.

Second, we empirically find that vote-k obviously underperform I^{2}CL, shown in Table 8 and Table 9. But this is not surprising because vote-k is still based on sentence-BERT similarity calculation, ignoring the internal triple semantics in test samples. For example, the improvements brought by vote-k become relatively obvious for GPT-4 compared to TableIE baseline. In text-davinci-3 results, with the same annotation budgets, vote-k improves the vanilla TableIE method by typically less than 0.5% absolute gain or is even worse than our random selection baseline (e.g., vote-k delivers around 36.42% (-0.94%), 42.17% (-0.73%) and 51.22% (+0.45%) on CoNLL04 with budget B=5, 15 and 25, respectively). The samples selected by vote-k lack consideration for entity and relation semantics, resulting in minimal advantage compared to directly sampling k samples for each relation type. Despite vote-k is worse than I^{2}CL in our setting, the purposes and settings of two methods are very different, and this finding does not necessarily demonstrate the superiority of I^{2}CL.

Table 8: Experimental results on CoNLL04 benchmark. Best results with different budgets and models are in **bold**.

Model	Method	CoNLL04		
		B=5	B=15	B=25
GPT-3	TextIE	19.85	32.83	40.35
	CodeIE	36.23	41.29	49.98
	TableIE	**37.36**	**42.90**	**50.77**
	+ vote-k	36.42 ↓	42.17 ↓	51.22 ↑
	+ I^{2}CL$_{\texttt{top-k}}$	25.97 ↓	39.49 ↓	47.71 ↓
	+ I^{2}CL$_{\texttt{balance}}$	**39.30** ↑	**46.77** ↑	54.97 ↑
	+ I^{2}CL$_{\texttt{coverage}}$	36.21 ↑	45.41 ↑	**55.36** ↑
ChatGPT	TextIE	23.55	37.53	42.95
	CodeIE	36.83	42.87	49.78
	TableIE	**37.29**	**43.15**	**50.31**
	+ vote-k	36.87 ↓	42.56 ↓	51.64 ↑
	+ I^{2}CL$_{\texttt{top-k}}$	27.70 ↓	43.56 ↑	49.72 ↓
	+ I^{2}CL$_{\texttt{balance}}$	**40.18** ↑	47.24 ↑	55.33 ↑
	+ I^{2}CL$_{\texttt{coverage}}$	36.35 ↑	**47.38** ↑	**56.45** ↑
GPT-4	TextIE	24.92	38.35	46.53
	CodeIE	36.77	43.25	50.19
	TableIE	**37.80**	**43.81**	**51.38**
	+ vote-k	37.22 ↓	45.16 ↑	53.25 ↑
	+ I^{2}CL$_{\texttt{top-k}}$	28.41 ↓	44.11 ↑	50.44 ↓
	+ I^{2}CL$_{\texttt{balance}}$	**40.51** ↑	**48.93** ↑	57.54 ↑
	+ I^{2}CL$_{\texttt{coverage}}$	37.63 ↑	48.70 ↑	**58.12** ↑

Table 9: Experimental results on NYT benchmark. Best results with different budgets and models are in **bold**. The results are all based on the `gpt-3.5-turbo-16k`.

Model	Method	NYT		
		B=24	B=48	B=72
ChatGPT	TextIE	18.85	18.88	19.44
	CodeIE	28.23	28.75	29.78
	TableIE	**29.31**	**29.84**	**30.45**
	+ vote-k	30.54 ↑	31.63 ↑	31.77 ↑
	+ I^{2}CL$_{\texttt{top-k}}$	**32.28** ↑	**34.92** ↑	**35.66** ↑
	+ I^{2}CL$_{\texttt{balance}}$	29.23 ↓	30.24 ↑	30.42 ↓
	+ I^{2}CL$_{\texttt{coverage}}$	31.44 ↑	32.21 ↑	32.89 ↑

C. Clarify of Balanced Strategy

In balanced strategy, we mention that this strategy possibly increases the annotation cost, especially under the adverse effects of imbalanced unlabeled data distribution in relation types. The potentially higher cost of annotation comes from some of the top sorted unlabeled samples may potentially belong to the same relation type. Suppose we aim to annotate 15 samples for 5 relations (i.e., 3 samples for each relation), but we may find that the top sorted 15 samples only contain 1 or 2 samples for a specific relation. Then we perhaps require the top sorted 20-30 samples to complete this balanced annotation. We call this "higher annotation cost" because even though we only annotate 15 samples in the end, we actually checked over 15 samples. But this potential for more checks will not appear in other two strategies.

D. Clarify of Term "Annotation"

The "annotation" actually means that the labels of selected samples are annotated by humans in practical scenarios, not LLMs. In the context of this paper, we assume that the labels in the annotated datasets are unknown (in fact, the samples have already been annotated and serve as benchmarks). We perform our TableIE and I^{2}CL methods on these "fake" unlabeled samples, then the selected unlabeled samples are "annotated" by aligning sentences with golden labels in original annotated datasets. In other words, we imitated the process of human annotation of data.

Acknowledgements

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.