C-ICL: Contrastive In-context Learning for Information Extraction

Ying Mo[1], **Jiahao Liu**[2], **Jian Yang**[1*], **Qifan Wang**[3], **Shun Zhang**[1], **Jingang Wang**[2], **Zhoujun Li**[1 *]
[1]Beihang University, Beijing, China
[2]Meituan, Beijing, China [3]Meta AI, New York, United States
{moying, jiaya, shunzhang, lizj}@buaa.edu.cn,
{liujiahao12, wangjingang02}@meituan.com, wqfcr@fb.com

Abstract

There has been increasing interest in exploring the capabilities of advanced large language models (LLMs) in the field of information extraction (IE), specifically focusing on tasks related to named entity recognition (NER) and relation extraction (RE). Although researchers are exploring the use of few-shot information extraction through in-context learning with LLMs, they tend to focus only on using correct or positive examples for demonstration, neglecting the potential value of incorporating incorrect or negative examples into the learning process. In this paper, we present C-ICL, a novel few-shot technique that leverages both correct and incorrect sample constructions to create in-context learning demonstrations. This approach enhances the ability of LLMs to extract entities and relations by utilizing prompts that incorporate not only the positive samples but also the reasoning behind them. This method allows for the identification and correction of potential interface errors. Specifically, our proposed method taps into the inherent contextual information and valuable information in hard negative samples and the nearest positive neighbors to the test and then applies the in-context learning demonstrations based on LLMs. Our experiments on various datasets indicate that C-ICL outperforms previous few-shot in-context learning methods, delivering substantial enhancements in performance across a broad spectrum of related tasks. These improvements are noteworthy, showcasing the versatility of our approach in miscellaneous scenarios.

1 Introduction

Information Extraction (IE) is an important task in natural language processing, which aims to obtain structured knowledge from plain text. It can be applied to different domains, such as knowledge graph construction (Zhong et al., 2023) and question answering (Aliod et al., 2006). With the rise of large language models (Brown et al., 2020; Min et al., 2022; Touvron et al., 2023; Achiam et al., 2023), information extraction has made significant progress (Li et al., 2023a; Xu et al., 2023).

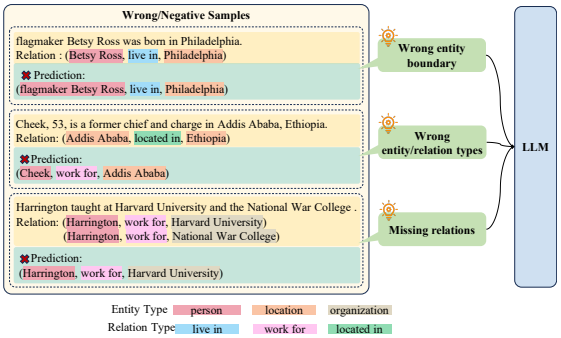


Figure 1: Illustration of our method C-ICL using wrong/negative samples. Take the RE task as an example to illustrate. Wrong/negative samples possess valuable information that LLMs may use to predict the type of IE errors, prompting the model to avoid or correct similar errors.

Recent advancements in few-shot IE have shifted focus from traditional supervised fine-tuning approaches towards in-context learning (ICL) demonstrations with large language models (LLMs) (Chen et al., 2023; Lyu et al., 2023). Prior works (Wei et al., 2023b; Chen et al., 2023; Ma et al., 2023; Wadhwa et al., 2023) have explored the use of natural language prompts or ICL demonstrations to guide LLMs in labeling test data under few-shot settings, sometimes requiring additional pre-training or fine-tuning steps. To align more closely with the structured nature of information extraction tasks, recent methods (Li et al., 2023b; Sainz et al., 2023; Wang et al., 2023a,b; Wan et al., 2023) adopt code-like or structured prompts to improve the consistency between pre-training and inference. *However, these methods can not fully unleash the potential of LLMs, partly due to the reliance of models on limited positive data and their inability*

* Corresponding author.

Findings of the Association for Computational Linguistics: EMNLP 2024, pages 10099–10114
November 12-16, 2024 ©2024 Association for Computational Linguistics

to learn from their own errors.

To address this, we propose a contrastive in-context learning approach that utilizes both correct (positive) and incorrect (negative) examples to enhance the learning process of LLMs by exposing them to a broader spectrum of scenarios, including typical mistakes as shown in Figure 1. This method aims to exploit the often-overlooked value in negative data, thus enabling more comprehensive and robust information extraction capabilities. Assume that the model has learned its own tasks and problem-solving modes from the positive IE data set, but the prediction is still wrong. Should it think about the reasons for the errors, summarize the types of reasons, and try to avoid the above problems in the subsequent inference? Then, adding negative sample-related information can help solve this problem. Inspired by this, we integrate right/positive and wrong/negative examples as ICL demonstrations to enhance the performance of in-context learning IE. Specifically, we first use a large model to generate the label of annotated data to select hard negative samples, then select positive samples semantically similar to the current test data from training data, and then design the most in-context demonstrations using different models (NL-LLMs or Code-LLMs). In the module that selects wrong/negative samples that contain more knowledge, we use semantic similarity-aware self-consistency to conduct ranking.

To demonstrate the superiority of our method, we conduct experiments on three NER and four RE benchmarks and carefully analyze the benefits of our approach. Our main contributions are summarized as follows:

- We develop contrastive in-context learning with both right/positive and wrong/negative instances demonstrations, simultaneously enhancing LLMs to extract valuable knowledge for information extraction.
- We select hard negative samples based on the effective retrieval strategy as in-context learning, which leverages to enhance the ability of information extraction.
- We conduct comprehensive experiments on benchmarks to demonstrate the performance of the proposed method, establishing new state-of-the-art results on most datasets.

2 Task Formulation

Given a sentence X with l tokens $x_1, x_2, \cdots, x_l$, IE tasks are to predict structured target Y (NER or RE in this paper) from x. The target Y in NER is entity spans $E((e,t)|x_i, \ldots, x_j)$ with entity types. the e is entity in the sequence, and t is the entity type in the predetermined entity types $\mathcal{T}$ (such as `LOC`, `PER`, `ORG`). In the RE task, the target Y is a set of relations within entities, usually expressed in the form of a triple (e_1, r, e_2). We not only predict the $r \in \mathcal{R}$, but also the entity types t_1, t_2 of e_1, e_2. $\mathcal{R}$ denotes the relation types (e.g., `Work For`, `Live In`, `Located In`). types of e_1, e_2 also should be predicted. $t \in \mathcal{T}$ means the entity type. We treat IE as a text generation task that completes the text to be predicted targets via LLMs. We define the task name as type instruction $\mathcal{I}$ to prompt the LLMs for NER or RE. When in NL-LLMs, its representation is a text sequence. In code-style prompts, it is treated as comments inspired by (Li et al., 2023b). We transform few-shot sample demonstrations into three parts: the first part is the sentence text, the second is the targets, and the third is a flag denoting whether the sample is positive or not. In NL-LLMs, the output is a list of tuples $[(e_1, t_1), \ldots, (e_j, t_j)]$. while in code generation, it is expressed as the operation of adding tuples to the dictionary like " entity_dict["person"].append["Steve"] ". The given test sentence has the first two parts, like the few-shot sample demonstrations. In our work, we pay more attention to code-style large model information extraction beacause of its structure.

3 C-ICL

3.1 Model Overview

We introduce C-ICL as shown in Figure 2, a novelty few-shot in-context learning method for information extraction, which predicts the right label via contrastive samples using LLMs. Unlike prior methods, which only use the samples with gold labels as the in-context learning demonstrations and ignore the knowledge in the wrong/negative samples predicted by models. Following the prior approaches (Li et al., 2023b; Guo et al., 2023b), our C-ICL uses code-style demonstrations for information extraction. It consists of building the contrastive samples (right/positive and wrong/negative samples) through retrieval strategies and prompts the LLMs' possible incorrect prediction problems.

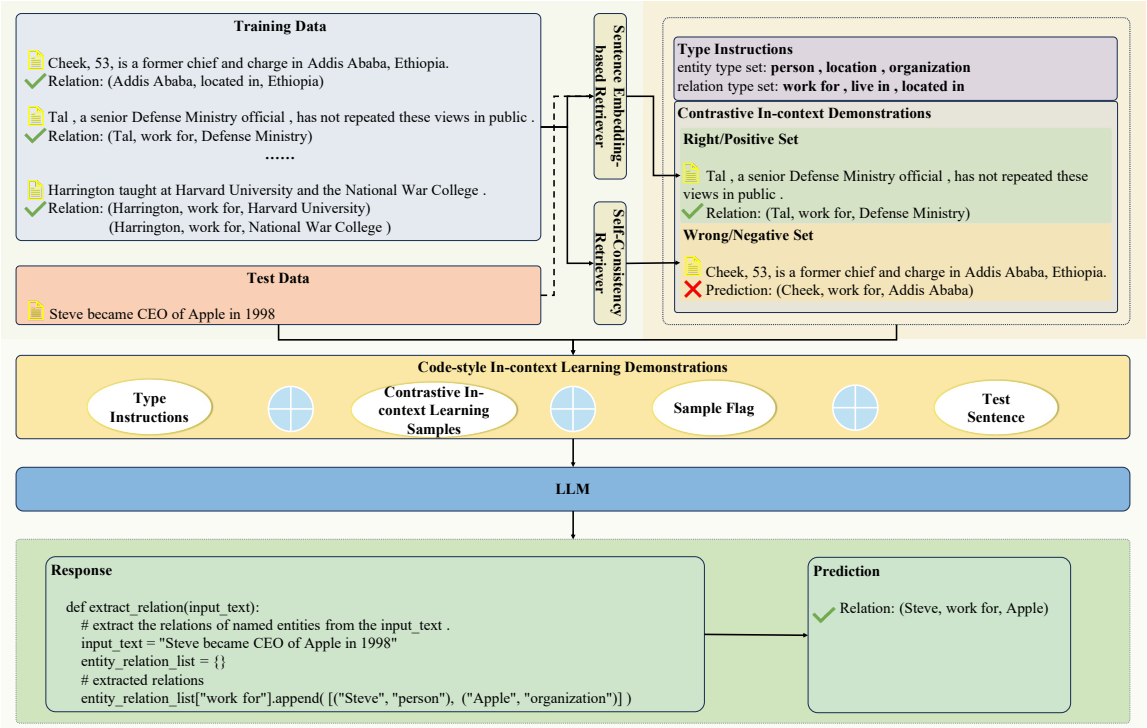


Figure 2: The overview of our method C-ICL for IE. Due to the structures of NER and RE tasks being similar, Take the RE task as an illustration to display the design in this figure. The predictions of the Wrong/Negative set are obtained through LLM. Right/Positive set and Wrong/Negative set are from the training or validation dataset.

3.2 LLM-based Information Extraction

Considering the natural language and structure of information extraction, we mainly choose code paragram style LLM to solve the problem. We take the query text as a variable and define the target of the IE task as a variable to return in the functions of the code program, which can illustrate the goal of IE functions. Beacause the given defined entity types and relation types have certainty and uniqueness, we think of it as a dictionary library. The type is the keyword in this dictionary, and the entity is the list element under the keyword.

$$\begin{aligned} y_{ner} &= \text{entity\_dict[t].append(e)} \\ y_{re} &= \text{entity\_relation\_dict[r].append(} \\ &\quad \text{[(e1, t1), (e2, t2)])} \end{aligned} \tag{1}$$

where y stands the expected output of the target Y in code-style LLMs. $entity\_dict$ and $entity\_relation\_dict$ are the return variables denoting the representations of the different target Y of NER and RE. e is the entity span that contains the tokens $x_i, \ldots, x_j$ in the given sentence X. t is one of the entity types $\mathcal{T}$ and r is one of the relation types $\mathcal{R}$. t, r denote the keys of the dictionary in the code function. $(e1, t1)$ and $(e2, t2)$ are the subject and object entities in a relation.

3.3 In-context Demonstrations Construction

We construct in-context demonstrations for each test sentence fed to the LLMs. Each demonstration consists of the following components.

Types Instructions To enhance the model's recognition of types in IE tasks, we provide predefined types ($\mathcal{R}$ or $\mathcal{T}$) in the demonstrations, which prompt LLMs for the purpose of the task, extracting information in a more targeted manner. We use comments to list possible predefined types instead of using code initialization to represent all types like (Guo et al., 2023b; Wang et al., 2023b). If there are too many types, this representation will increase the length of the in-context learning display. We put it at the front of the display to give a hint. In our method, The type instructions are shown the following way, illustrated as Figure 3:

$$\mathcal{D}_{\mathcal{T}} = \#\text{Given entity type set: } \mathcal{T}\,. \tag{2}$$

$$\begin{aligned} \mathcal{D}_{\mathcal{R}} = &\#\text{ Given entity type set: } \mathcal{T}; \\ &\#\text{Given relation type set: } \mathcal{R}\,. \end{aligned} \tag{3}$$

where $\mathcal{D}_{\mathcal{T}}$ means the entity type demonstration for NER task. $\mathcal{D}_{\mathcal{R}}$ denotes the relation type demonstration for the RE task. $\mathcal{T}$ likes "`LOC`, `PER`, `ORG`" and $\mathcal{R}$ is similar "`Work For`, `Live In`, `Located In`".

Type Instructions
#Given entity type set: **person , location , organization**
#Given relation type set: **work for , live in , located in**

Figure 3: An example of types instructions in-context demonstrations. Take the RE task as an illustration.

Right/Positive
```
def extract_relation(input_text):
    # extract the relations of named entities from the input_text .
    input_text = "Tal , a senior Defense Ministry official , has not repeated these views in public ."
    entity_relation_list = {}
    # extracted relations
    entity_relation_list["work for"].append([("Tal", "people"), ("Defense Ministry", "organization")])
```
#Above Result: Right.

Figure 4: An example of right/positive in-context demonstrations. Take the RE task as an illustration.

Contrastive Samples In this part, we introduce samples with two essences, one with golden labels and one with wrong labels. Positive samples (golden labels) can prompt the large model to generate text with what characteristics. Negative samples (wrong labels) are like a set of wrong questions, prompting problems that may occur in the model's inference process and avoiding them. The contrastive samples can work together to improve the ability for information extraction. We choose samples $\{(\hat{X}_i, \hat{Y}_i)\}_{i=1}^{\hat{n}}$ ($\hat{n}$ is the number of right/positive samples) close to the current test data through semantic similarity and represent them as samples with golden labels. Using the self-consistency method, we select hard negative samples $(\check{X}_i, \check{Y}_i)_{i=1}^{\check{n}}$ ($\check{n}$ means the count of wrong/negative samples) as samples with incorrect prediction results. Simultaneously, to prompt the large model to correct its errors when the prediction is wrong, we add the instructions (including the flag of whether the prediction is correct and the correct result) to the wrong/negative sample demonstration, as shown in Figure 4 and 5. The contrastive in-context demonstrations are the following:

$$\begin{aligned} \hat{\mathcal{D}} &= (\hat{X}_1 \oplus \hat{Y}_1) \oplus \cdots \oplus (\hat{X}n \oplus \hat{Y}n) \\ \check{\mathcal{D}} &= (\check{X}_1 \oplus \check{Y}_1) \oplus \cdots \oplus (\check{X}n \oplus \check{Y}n) \\ \mathcal{D} &= \hat{\mathcal{D}} \oplus \check{\mathcal{D}} \end{aligned} \tag{4}$$

where $\oplus$ means concatenation. $(X_i \oplus Y_i)$ means the presentation of the function of a sample in the code-style prompt. If the expression of wrong/negative samples is completely consistent with the positive sample, without training or fine-tuning, it is difficult for LLMs to recognize that this is a negative sample and its meaning. To better let negative samples guide the model for avoiding errors, we introduce a flag instruction behind the response to differentiate positive and negative samples. Furthermore, the flag instruction contains the right outputs for the given sample, as shown in Figure 4 and 5.

$$flag = \begin{cases} \text{"\#Above Result: Right."} & \text{if } X \text{ is } + \\ \text{"\#Above Result: Wrong. Right Result is blow: D"} & \text{if } X \text{ is } - \end{cases} \tag{5}$$

where $flag$ is the demonstration of judging positive and negative samples. D is the corrected representation of the incorrect answer. $+$ and $-$ denote positive and negative samples, respectively.

Wrong/Negative
```
def extract_relation(input_text):
    # extract the relations of named entities from the input_text .
    input_text = "Harrington taught at Harvard University and the National War College ."
    entity_relation_list = {}
    # extracted relations
    entity_relation_list["work for"].append([("Harrington", "people"), ("Harvard University", "organization")])
```
#Above Result: Wrong. Right Result is blow :
```
def extract_relation(input_text):
    # extract the relations of named entities from the input_text .
    input_text = "Harrington taught at Harvard University and the National War College ."
    entity_relation_list = {}
    # extracted relations
    entity_relation_list["work for"].append([("Harrington", "people"), ("Harvard University", "organization")])
    entity_relation_list["work for"].append([("Harrington", "people"), ("National War College", "organization")])
```

Figure 5: An example of wrong/negative in-context demonstrations. Take the RE task as an illustration.

Test Sentence
```
def extract_relation(input_text):
    # extract the relations of named entities from the input_text .
    input_text = "Steve became CEO of Apple in 1998"
    entity_relation_list = {}
    # extracted relations
```

Figure 6: An example of test sentence in-context demonstrations. Take the RE task as an illustration.

Test sentence Similar to the above demonstration, it only converts the test text shown in Figure 6 and entity/relation types into the code-style input of the contrastive sample. The final contrastive in-context demonstration is formulated as follows:

$$Y_{test} = \mathcal{M}(Y|\mathcal{I}, \mathcal{D}, X_{test}) \tag{6}$$

where X_{test} means the code-style transformation of the test sentence. $\mathcal{M}$ is the large language model like CodeLlama(Touvron et al., 2023). The RE output of Y_{test} is like a response module in Figure 2, and the NER task is similar.

3.4 In-context Example Retrieval Strategies

Sentence Embedding-based Retrieval Strategy Liu et al. (2022) indicates that in-context learning

demonstrations similar to the test data in semantic embedding may result in more reliable outcomes. So, the selection of samples is crucial for few-shot information extraction. In this part, we use the sentence embedding-based retrieve to select the samples from the training dataset for in-context learning demonstrations. Inspired by the previous works (Gutierrez et al., 2022; Liu et al., 2022; Guo et al., 2023b), we use the k nearest neighbors to retrieve sentences. After ranking the semantic similarity, we select the top-k samples with entities or relations. Our work employs the code LLMs to calculate the semantic similarity via cosine similarity.

Self-Consistency Retrieval Strategy For the wrong/negative samples, we select hard negative samples from the training dataset, which contain valuable knowledge and can better reflect the fuzzy performance of large language models in information extraction tasks. We use the large model first to get the prediction results of the training dataset. In this step, we apply self-consistency (Wang et al., 2023c) with votes to obtain predictions with high confidence. For each prediction, we determine whether the sample is a hard negative sample by calculating the F1 score, which indicates that the prediction is very close to the correct result. Through this method, we can get high-quality hard negative samples as the wrong/negative samples of contrastive in-context demonstrations.

4 Experiments

4.1 Datasets

RE Datasets For relation extraction, we evaluate on datasets CoNLL04 (Roth and Yih, 2004), ACE05-R (Walker and Consortium, 2005), NYT (Riedel et al., 2010) and SciERC (Luan et al., 2018). We follow Lu et al. (2022) to split all these datasets.
NER Datasets We evaluate our approach on the NER task with CoNLL03 (Sang and Meulder, 2003), ACE04(Doddington et al., 2004), and ACE05-E(Walker and Consortium, 2005). We split the datasets followed by the works (Li et al., 2020; Mo et al., 2023, 2024; Li et al., 2023b). Table 3 shows the dataset statistics in Appendix A.

4.2 Experiments Setting

For each IE task, we make a k-shot training set following the previous work (Li et al., 2023b), which samples k training samples for each entity or relation type via retrieval strategies. Since we have introduced wrong/negative samples, the settings are slightly different. We try to keep the overall sample number consistent for a fair comparison. In our experiments, we set the numbers of samples as 20, 14, and 14 for NER datasets CoNLL03, ACE04, and ACE05-E respectively, introduced in contrastive in-context learning demonstrations. The numbers of contrastive in-context learning samples for RE datasets CoNLL04, ACE05-R, NYT, and SciERC are 20, 12, 24, and 14. We use CodeLlama (Touvron et al., 2023) as the backbone and set the max token length to 8k and top_p to 0.7. Setting temperature in [0.3, 0.6, 0.9] is dependent on different datasets. When sampling hard negative samples, we query each sample to the model 3 times to acquire a suitable response. The experiment details are listed in Appendix A.

4.3 Evaluation

As in prior work (Lu et al., 2022; Li et al., 2023b), we use entity F1 score and relation strict F1 score as the evaluation metrics for NER and RE tasks, respectively. Note the relation strict F1 score for RE. A relation is correct only if the relation type, entity span, and entity type are all right. To ensure the accuracy of the results, we perform three runs with different random seeds for each experiment.

4.4 Comparison Methods

To demonstrate the effectiveness of our method, we adopt several models for comparison as follows.
UIE The method (Lu et al., 2022) is further pretrained from T5 on the structured datasets. It employs the textual structured extraction language to represent output structures.
InstructUIE The method (Wang et al., 2023a) improves UIE by constructing expert-written instructions for fine-tuning, enabling consistent modeling of IE tasks and capturing inter-task dependencies.
CodeIE The method (Li et al., 2023b) proposes to recast the structured output in the form of code instead of natural language, and utilize generative code-LLMs to perform IE tasks.
Code4UIE The method (Guo et al., 2023b) is a universal retrieval-augmented code generation framework, which utilizes Python classes to define schemas and uses ICL to generate codes that extract structural knowledge from texts.
CodeKGC The method (Bi et al., 2023) leverages the structural knowledge inherent in code and employs schema-aware prompts and rationale-enhanced generation to improve performance.

Model	Paradigm	Backbone	RE			
			CoNLL04	ACE05	NYT	SciERC
UIE (Lu et al., 2022)	SFT	T5-large	75.00	**66.06**	/	36.53
InstructUIE (Wang et al., 2023a)	SFT	Flan-T5-11B	**78.48**	/	**90.47**	**45.15**
CodeIE (Li et al., 2023b)	ICL	Code-davinci-002	53.10	14.02	32.17	7.74
Code4UIE (Guo et al., 2023b)	ICL	Gpt-3.5-turbo-16k	54.40	11.50	51.70	/
CodeKGC (Bi et al., 2023)	ICL	Text-davinci-003	49.80	/	/	**24.00**
C-ICL (ours)	ICL	CodeLlama-7B	53.27	18.75	58.39	12.13
	ICL	CodeLlama-13B	56.43	20.57	60.16	15.29
	ICL	CodeLlama-34B	**56.93**	**22.31**	**60.92**	17.33

Table 1: The experiment performances on RE benchmarks. SFT denotes the model adopts supervised fine-tuning with training data. ICL means the model uses in-context learning. Results are statistically significant with respect to baselines (p-value < 0.05).

Model	Paradigm	Backbone	NER		
			CoNLL03	ACE04	ACE05-E
UIE (Lu et al., 2022)	SFT	T5-large	**92.99**	**86.89**	85.78
InstructUIE (Wang et al., 2023a)	SFT	Flan-T5-11B	92.94	/	**86.66**
CodeIE (Li et al., 2023b)	ICL	Code-davinci-002	82.32	**55.29**	54.82
Code4UIE (Guo et al., 2023b)	ICL	Gpt-3.5-turbo-16k	79.70	54.0	**57.00**
Self-Improving (Xie et al., 2023b)	ICL	Gpt-3.5-turbo	83.51	/	55.54
C-ICL (ours)	ICL	CodeLlama-7B	83.98	47.88	45.65
	ICL	CodeLlama-13B	85.62	49.69	48.04
	ICL	CodeLlama-34B	**87.36**	54.47	55.65

Table 2: The experiment performances on NER benchmarks. SFT denotes the model adopts supervised fine-tuning with training data. ICL means the model uses in-context learning. Results are statistically significant with respect to baselines (p-value < 0.05).

Self-Improving The method (Xie et al., 2023b) uses the LLM to make predictions on the unlabeled data via self-consistency and explores strategies to select reliable annotations for NER task.

4.5 Results

RE Results Table 1 shows the results of the RE task. Among the datasets, NYT has the most significant improvement, whose results based on CodeLlama-7B exceed CodeIE and Code4UIE by +26.22% +6.69%, respectively. The results of NYT based on CodeLlama-34B exceed CodeIE and Code4UIE by +28.75% +9.22%, respectively. Our method gains an improvement of at least 4.73% in the F1 score compared to the LLM-based baselines for ACE05. Although CoNLL04 on CodeLlama-7B is slightly weaker than Code4UIE, there is a specific improvement on CodeLlama-13B and CodeLlama-34B by +2.03 points. At SciERC, our method improves by +4.39 points compared to CodeIE but is lower than CodeKGC. It is lower than CodeKGC because CodeKGC re-structure the corpus into code format and builds a new dataset for pre-training to effect rationale-enhanced generation. In addition, the backbone of CodeKGC is a larger model(Fu et al., 2023) than that we used.

NER Results Table 2 shows the results of the NER task. Our method achieves superior performance overall compared with the previous baselines, proving the effectiveness of our method in the information extraction subtask NER. We get an F1 score of 87.36% on the CoNLL03 dataset, increased by +3.76 points compared to the in-context learning methods CodeIE (Li et al., 2023b), Code4UIE(Guo et al., 2023b), and Self-Improving (Xie et al., 2023b). for ACE05-E, our model performs slightly better than the CodeIE method by 0.54 points. For ACE04, our results are weaker than those above LLM-based baselines. The improvement in performance of our method is not significant on these two datasets overall. The main reasons include that 1) these two datasets contain nested entities, and their error types are more numerous and complex compared to those of common NER tasks; 2) the added wrong/negative samples, which may lead to longer text length, affect the capture of contextual information, and lead to performance degradation; 3) The LLMs-based baselines are the GPT-based(Brown et al., 2020; OpenAI, 2022) methods, which are stronger pre-trained large language model. It should be noted that LLMs also have limitations with the nested

ACE04 and ACE05-E NER benchmarks during the experiments and enhance reasoning for complex situations. To further verify our method, we reproduce CodeIE and replace the backbone with CodeLlama. See Appendix B for relevant results.

Compared with the supervised baselines UIE(Lu et al., 2022) and InstructUIE (Wang et al., 2023a), our method performs worse than them, but it is approaching them on CoNLL03, showing that our method gives the model an excellent hint to enhance the reasoning ability on this task.

5 Further Analysis

5.1 Ablation Study

To demonstrate the effectiveness of the proposed method, we conduct an ablation study of our method. We run experiments on RE and NER datasets based on CodeLlama-7b, CodeLlama-13b, and CodeLlama-34b. The results are demonstrated in Figure 7. ① Ours, which is the final approach with the contrastive in-context objectives; ② w/o wrong/negative, which denotes that the method only adopts the right/positive samples as in-context demonstrations. From the ablation experiments, our method C-ICL outperforms different levels of improvement effects than ②, indicating that the contrastive in-context learning with right/positive and wrong/negative samples could prompt the LLMs to learn the positive and effective knowledge information. It can be seen that the larger the model, the better the effect for IE tasks.

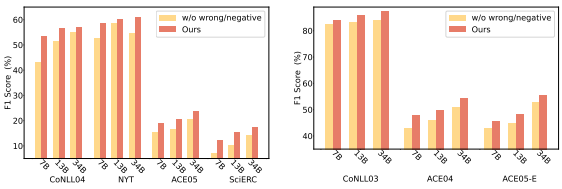


Figure 7: Alation study of C-ICL based on different CodeLlama for IE. Left and right figures denote the ablation study of RE and NER tasks, respectively.

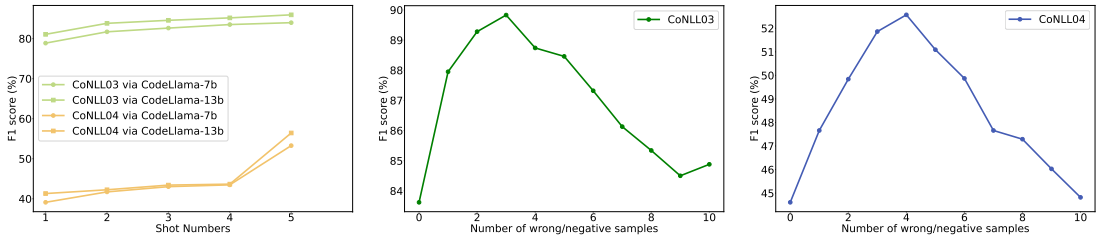
5.2 Examples of Contrastive ICL

Different Shot Numbers To illustrate the impact of different numbers of contrastive in-context demonstrations on information extraction, we conducted further experiments on the CoNLL03 (NER) and CoNLL04 (RE) via CodeLlama-7b and CodeLlama-13b. In this part, we ensure that the number of wrong/negative samples is consistent with two and that other display samples are right/-positive. The results are presented in Figure 8(a). As can be seen from it, the effect of each IE task increases as the number of shots increases. Overall, the effectiveness of each IE task tends to increase with the number of shots. This phenomenon is because as the number of sample instances increases, the large language models can glean more in-context information from them.

Proportion of Positive and Negative Demonstrations We conduct experiments to analyze the proportion of positive and negative samples in contrastive in-context demonstrations. In this part, we evaluate the CoNLL03 (NER) and CoNLL04 (RE) datasets and run experiments on the CodeLlama-7b. Note that we sample 300 test data for this analysis. We set the total number of demonstration samples to a particular value; the total number of contrastive samples is fixed, and the numbers of positive and negative samples are changing. The results are illustrated in Figure 8(b) and 8(c). Overall, the effectiveness of information extraction tends to rise first and then fluctuates with the increase in negative sample numbers. When positive samples provide adequate contextual information, adding negative samples can indirectly prompt the model to acquire other knowledge (types of errors that may occur) in entity or relation recognition and modify erroneous predictions. However, adding too many wrong/negative samples may increase noise, making the model ambiguous in identifying positive and negative knowledge. It will not serve as an optimistic prompt, resulting in poorer results. Besides, adding more negative samples makes the text longer in our design, which can lead to worse effects. Therefore, preferring a proper number of contrastive in-context demonstrations is necessary.

5.3 Retrieval Strategy of Contrastive ICL

We run the experiments of different retrieval strategies of contrastive in-context demonstrations in our method on CoNLL03 and NYT. The bars hatched in Figure 9 show the results for sampling the positive samples from training data. The sentence embedding-based retrieval strategy for positive samples performs better than random sampling positive data. In this strategy, the samples are more similar to the current test sequence in semantics. The bars without hatched in Figure 9 exhibit the results of different retrieval strategies for the wrong/negative samples. We find that 1) combining the self-consistency retrieval strategy and setting an



(a) (b) (c)

Figure 8: The performance of numbers of contrastive in-context demonstrations. (a) means the results of CoNLL03 (NER) and CoNLL04 (RE) with different shot numbers. (b) and (c) present the results of the proportion of positive and negative samples on CoNLL03 and CoNLL04, where we sample 300 test data for analysis.

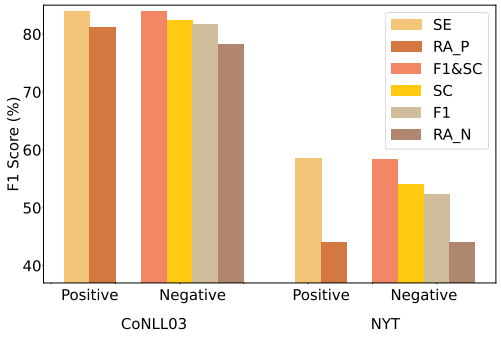


Figure 9: The comparison of retrieval strategies for contrastive samples. The bars with hatched and not show retrieval strategies for positive and negative samples, respectively. SE refers to the sentence embedding-based retrieval strategy, RA_$*$ to random sample, SC to the self-consistency strategy, F1 to retrieve sample by the F1 score, F1&SC to retrieve sample via F1 and SC.

F1 score threshold for retrieving hard negative samples can result in better performance. The strategy can obtain high-quality samples to prompt LLMs to learn from more knowledge. 2) Random retrieval for sampling wrong/negative data has lower effects than other strategies and it would cause fluctuation.

5.4 Case Study

We select some cases of typical test samples to illustrate better the amendment of our method in Figure 10. Given example 1, we show the more similar positive and negative samples as contrastive ICL demonstrations in the RE task. In this case, "southwestern France" and "west-central France" in the test sample are easily identified as "location" entities. The introduction of negative samples in ICL demonstrates potential issues that may arise during the test inference process. It further points out errors and provides corrective prompts, assisting the LLM model to accurately infer the correct results. We also discuss the case of the NER task. More cases can be seen in Appendix C.

[Type Instructions]
entity type set: **person , location , organization**
relation type set: **work for , live in , located in**

[Right/Positive]
```
def relation_extraction(input_text):
    """ extract the relations of named entities from the input_text . """
    input_text = "He and the others leased land in the desert west of this Shiite Moslem holy city about 65 miles south of Baghdad , Iraq 's capital."
    entity_relation_list = {}
    # extracted relations
    entity_relation_list["located in"].append([("Baghdad", "location"), ("Iraq", "location")])
#Above Result: Right.
```
.....

[Wrong/Negative]
```
def relation_extraction(input_text):
    """ extract the relations of named entities from the input_text . """
    input_text = "Thelma Ann Miller , 23 ; and Toby Miller , 15 , drowned Monday about 1 -LCB- miles northeast of Pinedale in rural western New Mexico , said Alex Ashley , a deputy state medical investigator."
    entity_relation_list = {}
    # extracted relations
    entity_relation_list[" located in "].append([(" Pinedale", "location"), ("western New Mexico", "location")])
#Above Result: Wrong. Right Result is blow :
def relation_extraction(input_text):
    """ extract the relations of named entities from the input_text . """
    input_text = "Thelma Ann Miller , 23 ; and Toby Miller , 15 , drowned Monday about 1 -LCB- miles northeast of Pinedale in rural western New Mexico , said Alex Ashley , a deputy state medical investigator."
    entity_relation_list = {}
    # extracted relations
    entity_relation_list["located in"].append([(" Pinedale", "location"), ("New Mexico", "location")])
```
.....

[Test Data]
```
def relation_extraction(input_text):
    """ extract the relations of named entities from the input_text . """
    input_text = "Urutigoechea and the others were arrested Wednesday in the cities of Bayonee and Bonloc in southwestern France in Poitiers in west-central France ."
    entity_relation_list = {}
    # extracted relations
```
[Output]
```
    entity_relation_list ["located in"].append([("Bayonee","location"), ("France" ,"location")])
    entity_relation_list ["located in"].append([(" Bonloc","location"), ("France" ,"location")])
    entity_relation_list ["located in"].append([(" Poitiers","location"), ("France" ,"location")])
```

Figure 10: Case study. The example stands for the results of RE. The red text means the incorrect labels and the green text means the right labels.

6 Related Work

Generative Information Extraction Due to the differences in subtasks (e.g. NER and RE), information extraction (IE) has seen many task-specific supervised models, including understanding models (Lample et al., 2016; Tang et al., 2018; Yu et al., 2020; Wang et al., 2018, 2020, 2019) and generation models (Yan et al., 2021; Lu et al., 2022; Mo et al., 2023), span-based models (Yu et al., 2020; Ye et al., 2022). Some Transformer-based models have been adapted for many NLP tasks, such as information extraction (Yang et al., 2022a; Mo et al., 2024; Wei et al., 2023a), intent discovery (Zhang et al., 2024b,a), and machine translation (Yang et al., 2022c,b, 2021a, 2023; Zhu et al., 2023). Through scaling up the model size, large language models (LLMs) produce competitive results without training by unifying different IE subtasks. NL-LLMs (Wei et al., 2023b) converts structured information tasks into natural language, but

there is an inconsistency between the reasoning goal and pre-training. Code-LLMs (Chai et al., 2023; Li et al., 2023b; Guo et al., 2023b; Bi et al., 2023) converts the text-to-structure IE task into a structure-to-structure code generation task.

In-context Learning In-context learning (ICL) can be enhanced in large-scale LLMs (Brown et al., 2020; Touvron et al., 2023; Bai et al., 2023; Guo et al., 2023a) by constructing valuable demonstrations (Yang et al., 2021b; Liu et al., 2022; Rubin et al., 2022; Dong et al., 2023; Wang et al., 2023d). The selected samples may have a positive or negative effect on in-context learning(Nguyen and Wong, 2023). Some researchers (Liu et al., 2022; Guo et al., 2023b) proposes to use *K*NN method to retrieve similar samples and Wan et al. (2023) employs task-aware retrieval and gold label-induced reasoning representation to select appropriate samples. Wei et al. (2023b); Xie et al. (2023a) use dialogue and question-answer methods. Code-style prompts convert them into program methods (Li et al., 2023b) or classes (Guo et al., 2023b; Bi et al., 2023).

Hard negative sample Hard negative samples should have different labels from the anchor sample but have embedding features very close to the anchor embedding. Different from learning and transferring knowledge through positive samples, models may obtain valuable information from negative samples to enhance the model performance (Robinson et al., 2021; Radenovic et al., 2023; Mo et al., 2024). Introducing negative samples can directly assist with positive samples to comprehensively extract helpful knowledge for LLM.

7 Conclusion

In this work, we introduce C-ICL, contrastive in-context learning for few-shot information extraction, including right/positive and wrong/negative demonstrations. In addition through type instruction demonstrations prompt mention tags in the IE task. From the contrastive samples, the LLMs could obtain effective information and indirect but positive, valuable additional knowledge for IE tasks. Besides, our method adopts semantic similarity retrieval strategies and self-consistency votes to retrieve in-context examples better suited for the current sentence and task, significantly improving IE performance. Extensive experiments prove the effectiveness of C-ICL on various benchmarks.

Limitations

We acknowledge the following limitations of this study: (1) This work focuses on exploring the in-context learning for few-shot NER and RE tasks. The investigation of this paradigm on other IE tasks has not been studied yet. (2) We apply the common sentence embedding similarity for retrieving positive samples. We use self-consistency and confidence F1 score to obtain hard negative samples as in-context demonstrations. There might be other diverse strategies for measuring suitable positive samples and the quality of hard negative samples. (3) The performance of our work still lags behind previous fully-supervised methods.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (Grant Nos. 62276017, U1636211, 61672081), and the State Key Laboratory of Complex & Critical Software Environment (Grant No. SKLSDE-2021ZX-18).

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Diego Mollá Aliod, Menno van Zaanen, and Daniel Smith. 2006. Named entity recognition for question answering. In *Proceedings of the Australasian Language Technology Workshop, ALTA 2006, Sydney, Australia, November 30-December 1, 2006*, pages 51–58. Australasian Language Technology Association.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Zhen Bi, Jing Chen, Yinuo Jiang, Feiyu Xiong, Wei Guo, Huajun Chen, and Ningyu Zhang. 2023. Codekgc: Code language model for generative knowledge graph construction. *CoRR*, abs/2304.09048.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Linzheng Chai, Dongling Xiao, Zhao Yan, Jian Yang, Liqun Yang, Qian-Wen Zhang, Yunbo Cao, and Zhoujun Li. 2023. QURG: question rewriting guided context-dependent text-to-sql semantic parsing. In *PRICAI 2023: Trends in Artificial Intelligence - 20th Pacific Rim International Conference on Artificial Intelligence, PRICAI 2023, Jakarta, Indonesia, November 15-19, 2023, Proceedings, Part II*, volume 14326 of *Lecture Notes in Computer Science*, pages 275–286. Springer.

Jiawei Chen, Yaojie Lu, Hongyu Lin, Jie Lou, Wei Jia, Dai Dai, Hua Wu, Boxi Cao, Xianpei Han, and Le Sun. 2023. Learning in-context learning for named entity recognition. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 13661–13675. Association for Computational Linguistics.

George R. Doddington, Alexis Mitchell, Mark A. Przybocki, Lance A. Ramshaw, Stephanie M. Strassel, and Ralph M. Weischedel. 2004. The automatic content extraction (ACE) program - tasks, data, and evaluation. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation, LREC 2004, May 26-28, 2004, Lisbon, Portugal*. European Language Resources Association.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, Lei Li, and Zhifang Sui. 2023. A survey on in-context learning. *Preprint*, arXiv:2301.00234.

Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. 2023. Complexity-based prompting for multi-step reasoning. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Hongcheng Guo, Jian Yang, Jiaheng Liu, Liqun Yang, Linzheng Chai, Jiaqi Bai, Junran Peng, Xiaorong Hu, Chao Chen, Dongfeng Zhang, Xu Shi, Tieqiao Zheng, Liangfan Zheng, Bo Zhang, Ke Xu, and Zhoujun Li. 2023a. Owl: A large language model for it operations. *arXiv preprint arXiv:2309.09298*.

Yucan Guo, Zixuan Li, Xiaolong Jin, Yantao Liu, Yutao Zeng, Wenxuan Liu, Xiang Li, Pan Yang, Long Bai, Jiafeng Guo, and Xueqi Cheng. 2023b. Retrieval-augmented code generation for universal information extraction. *CoRR*, abs/2311.02962.

Bernal Jimenez Gutierrez, Nikolas McNeal, Clayton Washington, You Chen, Lang Li, Huan Sun, and Yu Su. 2022. Thinking about GPT-3 in-context learning for biomedical ie? think again. In *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 4497–4512. Association for Computational Linguistics.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *NAACL HLT 2016*, pages 260–270.

Bo Li, Gexiang Fang, Yang Yang, Quansen Wang, Wei Ye, Wen Zhao, and Shikun Zhang. 2023a. Evaluating chatgpt's information extraction capabilities: An assessment of performance, explainability, calibration, and faithfulness. *CoRR*, abs/2304.11633.

Peng Li, Tianxiang Sun, Qiong Tang, Hang Yan, Yuanbin Wu, Xuanjing Huang, and Xipeng Qiu. 2023b. Codeie: Large code generation models are better few-shot information extractors. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 15339–15353. Association for Computational Linguistics.

Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. 2020. A unified MRC framework for named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 5849–5859. Association for Computational Linguistics.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for gpt-3? In *Proceedings of Deep Learning Inside Out: The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures, DeeLIO@ACL 2022, Dublin, Ireland and Online, May 27, 2022*, pages 100–114. Association for Computational Linguistics.

Yaojie Lu, Qing Liu, Dai Dai, Xinyan Xiao, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. 2022. Unified structure generation for universal information extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 5755–5772. Association for Computational Linguistics.

Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 -*

November 4, 2018, pages 3219–3232. Association for Computational Linguistics.

Xinxi Lyu, Sewon Min, Iz Beltagy, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. Z-ICL: zero-shot in-context learning with pseudo-demonstrations. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 2304–2317. Association for Computational Linguistics.

Xilai Ma, Jing Li, and Min Zhang. 2023. Chain of thought with explicit evidence reasoning for few-shot relation extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 2334–2352. Association for Computational Linguistics.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 11048–11064. Association for Computational Linguistics.

Ying Mo, Hongyin Tang, Jiahao Liu, Qifan Wang, Zenglin Xu, Jingang Wang, Wei Wu, and Zhoujun Li. 2023. Multi-task transformer with relation-attention and type-attention for named entity recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP 2023, Rhodes Island, Greece, June 4-10, 2023*, pages 1–5. IEEE.

Ying Mo, Jian Yang, Jiahao Liu, Qifan Wang, Ruoyu Chen, Jingang Wang, and Zhoujun Li. 2024. MCL-NER: cross-lingual named entity recognition via multi-view contrastive learning. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pages 18789–18797.

Tai Nguyen and Eric Wong. 2023. In-context example selection with influences. *CoRR*, abs/2302.11042.

OpenAI. 2022. Chatgpt: Optimizing language models for dialogue. *OpenAI Blog*.

Filip Radenovic, Abhimanyu Dubey, Abhishek Kadian, Todor Mihaylov, Simon Vandenhende, Yash Patel, Yi Wen, Vignesh Ramanathan, and Dhruv Mahajan. 2023. Filtering, distillation, and hard negatives for vision-language pre-training. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 6967–6977. IEEE.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Machine Learning and Knowledge Discovery in Databases, European Conference, ECML PKDD 2010, Barcelona, Spain, September 20-24, 2010, Proceedings, Part III*, volume 6323 of *Lecture Notes in Computer Science*, pages 148–163. Springer.

Joshua David Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. 2021. Contrastive learning with hard negative samples. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Dan Roth and Wen-tau Yih. 2004. A linear programming formulation for global inference in natural language tasks. In *Proceedings of the Eighth Conference on Computational Natural Language Learning, CoNLL 2004, Held in cooperation with HLT-NAACL 2004, Boston, Massachusetts, USA, May 6-7, 2004*, pages 1–8. ACL.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning to retrieve prompts for in-context learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 2655–2671. Association for Computational Linguistics.

Oscar Sainz, Iker García-Ferrero, Rodrigo Agerri, Oier Lopez de Lacalle, German Rigau, and Eneko Agirre. 2023. Gollie: Annotation guidelines improve zero-shot information-extraction. *CoRR*, abs/2310.03668.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning, CoNLL 2003, Held in cooperation with HLT-NAACL 2003, Edmonton, Canada, May 31 - June 1, 2003*, pages 142–147. ACL.

Buzhou Tang, Jianglu Hu, Xiaolong Wang, and Qingcai Chen. 2018. Recognizing continuous and discontinuous adverse drug reaction mentions from social media using lstm-crf. *Wireless Communications and Mobile Computing*, 2018.

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

Somin Wadhwa, Silvio Amir, and Byron C. Wallace. 2023. Revisiting relation extraction in the era of large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 15566–15589. Association for Computational Linguistics.

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

Zhen Wan, Fei Cheng, Zhuoyuan Mao, Qianying Liu, Haiyue Song, Jiwei Li, and Sadao Kurohashi. 2023. GPT-RE: in-context learning for relation extraction using large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 3534–3547. Association for Computational Linguistics.

Xiao Wang, Weikang Zhou, Can Zu, Han Xia, Tianze Chen, Yuansen Zhang, Rui Zheng, Junjie Ye, Qi Zhang, Tao Gui, et al. 2023a. Instructuie: Multi-task instruction tuning for unified information extraction. *arXiv preprint arXiv:2304.08085*.

Xingyao Wang, Sha Li, and Heng Ji. 2023b. Code4struct: Code generation for few-shot event structure prediction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 3640–3663. Association for Computational Linguistics.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023c. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Yunli Wang, Zhiqiang Wang, Jian Yang, Shiyang Wen, Dongying Kong, Han Li, and Kun Gai. 2023d. Adaptive neural ranking framework: Toward maximized business goal for cascade ranking systems. *arXiv preprint arXiv:2310.10462*.

Yunli Wang, Yu Wu, Lili Mou, Zhoujun Li, and Wenhan Chao. 2019. Harnessing pre-trained neural networks with rules for formality style transfer. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3573–3578.

Yunli Wang, Yu Wu, Lili Mou, Zhoujun Li, and Wenhan Chao. 2020. Formality style transfer with shared latent space. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2236–2249.

Yunli Wang, Zhao Yan, Zhoujun Li, and Wenhan Chao. 2018. Response selection of multi-turn conversation with deep neural networks. In *Natural Language Processing and Chinese Computing: 7th CCF International Conference, NLPCC 2018, Hohhot, China, August 26–30, 2018, Proceedings, Part I 7*, pages 110–119. Springer.

Xiang Wei, Xingyu Cui, Ning Cheng, Xiaobin Wang, Xin Zhang, Shen Huang, Pengjun Xie, Jinan Xu, Yufeng Chen, Meishan Zhang, Yong Jiang, and Wenjuan Han. 2023a. Zero-shot information extraction via chatting with chatgpt. *CoRR*, abs/2302.10205.

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

Tingyu Xie, Qi Li, Jian Zhang, Yan Zhang, Zuozhu Liu, and Hongwei Wang. 2023a. Empirical study of zero-shot NER with chatgpt. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 7935–7956. Association for Computational Linguistics.

Tingyu Xie, Qi Li, Yan Zhang, Zuozhu Liu, and Hongwei Wang. 2023b. Self-improving for zero-shot named entity recognition with large language models. *CoRR*, abs/2311.08921.

Derong Xu, Wei Chen, Wenjun Peng, Chao Zhang, Tong Xu, Xiangyu Zhao, Xian Wu, Yefeng Zheng, and Enhong Chen. 2023. Large language models for generative information extraction: A survey. *CoRR*, abs/2312.17617.

Hang Yan, Tao Gui, Junqi Dai, Qipeng Guo, Zheng Zhang, and Xipeng Qiu. 2021. A unified generative framework for various NER subtasks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 5808–5822. Association for Computational Linguistics.

Jian Yang, Shaohan Huang, Shuming Ma, Yuwei Yin, Li Dong, Dongdong Zhang, Hongcheng Guo, Zhoujun Li, and Furu Wei. 2022a. CROP: zero-shot cross-lingual named entity recognition with multilingual labeled sequence translation. In *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 486–496. Association for Computational Linguistics.

Jian Yang, Shuming Ma, Li Dong, Shaohan Huang, Haoyang Huang, Yuwei Yin, Dongdong Zhang, Liqun Yang, Furu Wei, and Zhoujun Li. 2023. Ganlm: Encoder-decoder pre-training with an auxiliary discriminator. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 9394–9412. Association for Computational Linguistics.

Jian Yang, Shuming Ma, Haoyang Huang, Dongdong Zhang, Li Dong, Shaohan Huang, Alexandre Muzio, Saksham Singhal, Hany Hassan, Xia Song, and Furu Wei. 2021a. Multilingual machine translation systems from microsoft for WMT21 shared task. In *WMT 2021*, pages 446–455. Association for Computational Linguistics.

Jian Yang, Juncheng Wan, Shuming Ma, Haoyang Huang, Dongdong Zhang, Yong Yu, Zhoujun Li,

and Furu Wei. 2021b. Learning to select relevant knowledge for neural machine translation. In *Natural Language Processing and Chinese Computing - 10th CCF International Conference, NLPCC 2021, Qingdao, China, October 13-17, 2021, Proceedings, Part I*, volume 13028 of *Lecture Notes in Computer Science*, pages 79–91. Springer.

Jian Yang, Yuwei Yin, Shuming Ma, Dongdong Zhang, Zhoujun Li, and Furu Wei. 2022b. High-resource language-specific training for multilingual neural machine translation. In *IJCAI 2022*, pages 4461–4467.

Jian Yang, Yuwei Yin, Shuming Ma, Dongdong Zhang, Shuangzhi Wu, Hongcheng Guo, Zhoujun Li, and Furu Wei. 2022c. UM4: unified multilingual multiple teacher-student model for zero-resource neural machine translation. In *IJCAI 2022*, pages 4454–4460.

Deming Ye, Yankai Lin, Peng Li, and Maosong Sun. 2022. Packed levitated marker for entity and relation extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 4904–4917. Association for Computational Linguistics.

Juntao Yu, Bernd Bohnet, and Massimo Poesio. 2020. Named entity recognition as dependency parsing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 6470–6476. Association for Computational Linguistics.

Shun Zhang, Chaoran Yan, Jian Yang, Jiaheng Liu, Ying Mo, Jiaqi Bai, Tongliang Li, and Zhoujun Li. 2024a. Towards real-world scenario: Imbalanced new intent discovery. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 3949–3963.

Shun Zhang, Chaoran Yan, Jian Yang, Wei Zhang, Changyu Ren, Tongliang Li, Jiaqi Bai, and Zhoujun Li. 2024b. Tinid: A transfer and interpretable llm-enhanced framework for new intent discovery. In *Machine Learning and Knowledge Discovery in Databases. Research Track - European Conference, ECML PKDD 2024, Vilnius, Lithuania, September 9-13, 2024, Proceedings, Part V*, volume 14945, pages 195–212.

Lingfeng Zhong, Jia Wu, Qian Li, Hao Peng, and Xindong Wu. 2023. A comprehensive survey on automatic knowledge graph construction. *CoRR*, abs/2302.05019.

Wenhao Zhu, Hongyi Liu, Qingxiu Dong, Jingjing Xu, Lingpeng Kong, Jiajun Chen, Lei Li, and Shujian Huang. 2023. Multilingual machine translation with large language models: Empirical results and analysis. *CoRR*, abs/2304.04675.

A Implementation Experiment

A.1 Dataset Statistics

To ensure a comprehensive evaluation, we select a diverse set of datasets on the NER and RE tasks, including three NER benchmarks and four RE benchmarks. The specific statistics of these datasets, including the number of entity and relation types, as well as the distribution of instances across training, development, and test sets, are summarized in Table 3. This detailed breakdown provides insights into the dataset composition and serves as a reference for the robustness of the evaluation framework applied to our approach.

Datasets		Entity Types	Relation Types	Train	Dev	Test
NER	CoNLL03	4	/	14041	3250	3453
	ACE04	7	/	6202	745	812
	ACE05-E	7	/	7299	971	1060
RE	CoNLL04	4	5	922	231	288
	ACE05	7	6	10051	2420	2050
	NYT	3	24	56196	5000	5000
	SciERC	6	7	1861	275	551

Table 3: Statistics of NER and RE Datasets.

A.2 Implementation Experiment Details

We run all experiments with the deep learning framework PyTorch NVIDIA Tesla A100 GPUs. The specific configurations and hyperparameters used in our method are meticulously chosen to optimize performance. These parameters include the maximum sequence length, the batch size, the number of beams for beam search, the top-p, and temperatures for controlling the randomness of the output. The parameters are detailed in Table 4.

Parameters	Values
Max Sequence Length	8192
Batch Size	[1, 2]
Num_beams	1
Do_sample	True
Top_p	0.7
Temperature	[0.3,0.6,0.9]

Table 4: The main parameters of our method C-ICL based on CodeLlama.

B Supplementary Analysis

B.1 C-ICL vs CodeIE based on CodeLlama

Since ChatGPT (OpenAI, 2022) is a robust model, our method, which is based on the open-source model CodeLlama, may be at a disadvantage. To better illustrate the effectiveness of our method C-ICL, we reproduce CodeIE and replace its backbone with CodeLlama, and the relevant parameter settings are the same as ours. The results are shown in Table 5 and 6. From the tables, under the same model and parameters, we can see that our method is superior to CodeIE. It shows that our method with wrong/negative samples can provide more effective information to large models to improve the performance of IE.

Model	Backbone	NER		
		CoNLL03	ACE04	ACE05-E
CodeIE (Li et al., 2023b)	Code-davinci-002	82.32	55.29	54.82
	CodeLlama-7B	72.33	36.21	35.18
	CodeLlama-13B	79.30	38.82	35.87
	CodeLlama-34B	82.53	46.38	46.46
C-ICL (ours)	CodeLlama-7B	83.98	47.88	45.65
	CodeLlama-13B	85.94	49.69	48.04
	CodeLlama-34B	87.36	54.47	55.65

Table 5: The experiment performances of C-ICL and CodeIE via CodeLlama on NER benchmarks.

Model	Backbone	RE			
		CoNLL04	ACE05	NYT	SciERC
CodeIE (Li et al., 2023b)	Code-davinci-002	53.10	14.02	32.17	7.74
	CodeLlama-7B	29.43	7.89	29.04	9.64
	CodeLlama-13B	33.91	8.11	31.75	12.96
	CodeLlama-34B	36.03	15.94	34.92	10.55
C-ICL (ours)	CodeLlama-7B	53.27	18.75	59.68	12.13
	CodeLlama-13B	56.43	20.57	60.16	15.29
	CodeLlama-34B	56.93	23.49	60.92	17.33

Table 6: The experiment performances of C-ICL and CodeIE via CodeLlama on RE benchmarks.

B.2 Prediction Error

In order to explore the impact of wrong/negative samples as in-context demonstrations on the types of errors that may occur in information extraction, we conducted analysis experiments on CoNLL03, ACE04, CoNLL04, and SciERC datasets via the backbone CodeLlama-7B. The results are shown in the Table 7 and 8. The number of entities involved in the evaluation of CoNLL03 and ACE04 datasets are 5648 and 3035 respectively. The number of relations involved in the evaluation of CoNLL04 and SciERC datasets are 422 and 974 respectively.

C Supplementary Case Study

In this section, we present other examples of NER and RE test datasets in our experiments, as shown in Figure 11 In example 2, common issues that arise include the omission of entities or relations. By providing hints through negative samples, it is ensured that problems occurring in the generation process of test samples can be revised and corrected.

In Example 3, even with the provision of semantically similar positive and negative samples as ICL

Model	Backbone	Entity Type Error		Entity Span Error	
		CoNLL03	ACE04	CoNLL03	ACE04
CodeIE (Li et al., 2023b)	CodeLlama-7B	210	329	2300	2303
C-ICL (ours)	CodeLlama-7B	126	434	1842	2549

Table 7: Predicton errors on NER datasets. "Ent Type Error" means the predicted entity type of the entity is not in the predefined type set. "Ent Span Error" means the predicted entity span of the entity is not in the test text.

Model	Backbone	Entity Type Error		Entity Span Error		Relation Type Error	
		CoNLL04	SciERC	CoNLL04	SciERC	CoNLL04	SciERC
CodeIE (Li et al., 2023b)	CodeLlama-7B	13	93	492	1481	16	38
C-ICL (ours)	CodeLlama-7B	7	88	273	1233	7	17

Table 8: Predicton errors on RE datasets. "Ent Type Error" means the predicted entity type of the entity is not in the predefined type set. "Ent Span Error" means the predicted span of the entity is not in the test text. "Relation Type Error" means the predicted label is not in the predefined relation type set.

demonstrations, the correct output results are still not guaranteed. In this case, it can be observed that the semantics of the test data itself are challenging. “there” means a location and “highway” is a facility in the location “there”. The large model has not yet been handled perfectly to recognize such issues, even with the guidance provided by rich ICL demonstrations. For this situation, the need for further advancements in training and fine-tuning techniques to improve LLM’s interpretive capabilities and proficiency in handling complex inferences.

(a) Example 2

(b) Example 3

Figure 11: Supplementary case study of contrastive in-context learning. The red text means the incorrect labels and the green text means the right labels. Figure 11(a) means the results of NER. Figure 11(b) means the results of RE.