REA: Refine-Estimate-Answer Prompting for Zero-Shot Relation Extraction

Amirhossein Layegh, Amir H. Payberah, and Mihhail Matskin

KTH Royal Institute of Technology, Stockholm, Sweden
{amlk,payberah,misha}@kth.se

Abstract. Zero-shot relation extraction (RE) presents the challenge of identifying entity relationships from text without training on those specific relations. Despite significant advancements in natural language processing by applying large language models (LLMs), their application to zero-shot RE remains less effective compared to traditional models that fine-tune smaller pre-trained language models. This limitation is attributed to insufficient prompting strategies that fail to leverage the full capabilities of LLMs for zero-shot RE, considering the intrinsic complexities of the RE task. A compelling question is whether LLMs can address complex tasks, such as RE, by decomposing them into more straightforward, distinct tasks that are easier to manage and solve individually. We propose the *Refine-Estimate-Answer (REA)* approach to answer this question. This multi-stage prompting strategy of REA decomposes the RE task into more manageable subtasks and applies iterative refinement to guide LLMs through the complex reasoning required for accurate RE. Our research validates the effectiveness of REA through comprehensive testing across multiple public RE datasets, demonstrating marked improvements over existing LLM-based frameworks. Experimental results on the FewRel, Wiki-ZSL, and TACRED datasets show that our proposed approach significantly boosts the vanilla prompting F1 scores by 31.57, 19.52, and 15.39, respectively, thereby outperforming the performance of state-of-the-art LLM-based methods.

Keywords: Relation Extraction · Large Language Models · Prompting Strategy.

1 Introduction

Relation Extraction (RE) aims to identify and classify semantic relationships between entities in unstructured text [3]. RE has received significant attention in natural language processing (NLP) due to its pivotal role across various downstream tasks, including information retrieval [13], question-answering [22], and knowledge graph construction [24]. Despite extensive research, state-of-the-art solutions still face challenges, such as adaptability to new domains and generalization to unseen relations due to their reliance on annotated data. This reliance makes these solutions impractical for scenarios where data is scarce or costly

to obtain. Therefore, *zero-shot RE* [34], where no annotated data for unseen relations is available, has become a crucial yet complex problem to address [4].

Recent advancements in NLP, particularly the emergence of large language models (LLMs) like GPT-3 [2], have further revolutionized the landscape of NLP tasks, such as RE. Trained on vast amounts of diverse textual data, LLMs exhibit remarkable capabilities in understanding text and generating human-like responses. As a result, generative zero-shot RE, where LLMs are prompted to directly extract and generate relationships between entities from text without fine-tuning, has gained significant attention [38]. It is crucial to note that the design of the prompt plays a vital role in the performance of LLMs. Varied prompts for the same tasks can lead to considerable discrepancies in model outputs [1].

There are several strategies for prompting Large Language Models (LLMs) for generative zero-shot Relation Extraction (RE), including "*vanilla*" prompting—a term we use to describe the most straightforward form of prompting—, *in-context learning* [2], and *chain-of-thought* (CoT) [36]. However, these prompting strategies have limitations. For instance, vanilla prompting is considered ineffective despite its simplicity as it necessitates LLMs to perform non-trivial reasoning processes within a single step [18]. In-context learning, while promising, heavily relies on the careful selection and variation of in-context examples and prompt templates [20]. CoT prompting, aimed at providing additional context through intermediate reasoning steps, frequently struggles to generalize and solve problems more challenging than the in-context CoT examples, limiting its utility in scenarios such as zero-shot RE, which demands robust generalization capabilities [44].

A recent body of work exhibited remarkable progress on generative zero-shot RE by employing LLMs. For instance, QA4RE [41] reformulates RE as a multiple-choice question-answering (QA) task to align the RE with QA tasks. SUMASK [18] integrates a CoT approach and proposes summarize-to-ask prompting with an uncertainty estimator component to tackle the challenge of ensuring the reliability of LLM responses. Nonetheless, when measured against state-of-the-art zero-shot RE methods that leverage fine-tuning on smaller pre-trained language models (PLMs), these approaches tend to underperform. [7].

The inherent complexity of RE stems from the need to understand the semantics of entities, identify their types, capture the semantics embedded within relation labels, and align these semantics properly [3]. Consequently, we hypothesize that the current limitations in existing generative zero-shot RE frameworks might be attributed to the insufficiently sophisticated prompting strategies that fail to capture these complexities. These limitations hinder their ability to effectively guide LLMs through the essential reasoning processes required for RE tasks. This observation motivates us to investigate whether decomposing RE into more manageable subtasks, each aligned with the core complexities of RE, can enhance performance.

Moreover, recent research has highlighted the benefits of enabling LLMs to refine their initial responses through self-critique [23], leading to enhanced rea-

soning capabilities [8, 26]. Inspired by this, our proposed approach incorporates a similar concept via confidence elicitation, examining the impact of iterative refinement on model performance. Confidence elicitation allows the model to express its certainty in its predictions, providing valuable feedback for further refinement [15].

Drawing on these insights, we introduce *Refine-Estimate-Answer (REA)*, a multi-stage prompting strategy that merges decomposition with iterative refinement. This approach aims to significantly elevate the performance of generative zero-shot RE by exploiting LLMs capabilities without relying on external knowledge or additional components. This paper delves into the detailed methodology of REA prompting and explores its effectiveness in enhancing generative zero-shot RE tasks. Our key contributions are as follows:

- Development of REA, a novel multi-stage prompting approach designed to enhance generative zero-shot RE tasks.
- Comprehensive experiments and evaluations conducted on three publicly available RE datasets to assess the effectiveness of REA. The results demonstrate REA's superiority over existing generative zero-shot RE frameworks, outperforming vanilla prompting performance by 5.88 - 39.47 in the F1-score.
- Demonstrating that decomposing RE into manageable subtasks, aligned with the core complexities of RE, significantly improves the performance of generative zero-shot RE models. This finding suggests that the REA decomposition approach effectively addresses the inherent challenges of RE, leading to more accurate results.
- Exhibiting that allows LLMs to assess and refine their initial responses using confidence elicitation iteratively, REA achieves a measurable improvement in accuracy.

2 Related Work

Zero-Shot Relation Extraction Early efforts at RE predominantly explored diverse neural architectures such as RNN [6] and BiLSTM [32], showcasing the promise of neural models in identifying relationships. However, these approaches require substantial annotated corpora and do not perform satisfactorily in zero-shot scenarios. Introducing PLMs has led to notable progress in RE, leveraging transformer-based models to identify the relationships between entities [9, 35]. Specifically, RE-Matching [43] introduces a fine-grained semantic matching technique that refines using PLMs for zero-shot RE by distinctively handling entity and context correlations. Subsequently, the paradigm of prompt-tuning PLMs emerged as a solution to bridge the gap between pre-training and fine-tuning objectives to enhance the performance of PLMs in low-resource tasks [21]. In this regard, RelationPrompt [5] prompts PLMs to generate synthetic training examples, articulating specific relations. This generated dataset subsequently trains another PLM to perform zero-shot RE. Despite these advancements, fine-tuning and prompt-tuning PLMs often face challenges in generalization, necessitating additional tuning on annotated datasets to predict unseen relations in zero-shot settings accurately.

LLMs for Generative Relation Extraction Adopting LLMs demonstrating proficiency in various downstream tasks without necessitating any form of training or fine-tuning emerged as an effective strategy. Specifically, QA4RE [41] introduced a framework for zero-shot RE by adapting RE tasks into a multiple-choice question-answering problem. Similarly, ChatIE [37] employs ChatGPT [25] for zero-shot information extraction, transforming the task into a multi-step question-answering process. Moreover, SUMASK [18] presents a multi-stage zero-shot RE framework, integrating LLMs with a natural language inference module for uncertainty estimation. Despite outperforming other LLM-based methods, SUMASK complexity arises from generating multiple summaries, questions, and answers for each relation label. Additionally, its dependency on an external module for uncertainty estimation poses integration challenges in real-world scenarios.

Decomposing Approaches in LLMs Recent research, inspired by CoT prompting [36], has shown that LLMs can handle complex problems better by breaking them down into intermediate steps [14, 39]. This decomposition facilitates LLMs to clarify their reasoning by prompting them to generate intermediate rationales for their solutions [30, 44]. The power of decomposition extends beyond CoT, proving valuable in addressing various challenges associated with LLMs [8,11,40]. For instance, least-to-most prompting [44] breaks down a complex task into a series of simpler subtasks and then solves them sequentially. Chain-of-Verification (CoV) [11] adopts a step-wise breakdown in question answering tasks, which involves generating initial answers, formulating and answering verification questions, and refining the original answers. This method mitigates hallucination, prevents factual errors, and ultimately enhances accuracy. In the context of RE, the SUMASK [18] framework also leverages a decomposition strategy to enhance performance through an external uncertainty estimation component. This highlights the untapped potential of decomposition approaches in further advancing generative RE with LLMs.

3 Background

Relation Extraction (RE) aims to identify and classify the relationships between head and tail entities in a sentence (Figure 1(a)). Typically, examples in RE datasets are represented as pairs $(\mathbf{X}, \mathbf{Y})$, where $\mathbf{X} = \{x_1, x_2, \cdots, x_h, \cdots, x_t, \cdots, x_n\}$ denotes the input sentence with n tokens and x_h and x_t represent the head and tail entities, respectively, and $\mathbf{Y}$ denotes the corresponding relation label between the entity pair (x_h, x_t). Notably, $\mathbf{Y}$ belongs to a pre-defined set of labels $\mathbf{L}$ that include relation labels, such as `occupant`, `lives in`, and `head of government`. For example, given $\mathbf{X}$ = `{Claude Malhuret is the mayor of Vichy, France.}`, x_h = `{Claude Malhuret}`, and x_t = `{Vichy}`, the relation label $\mathbf{Y}$ would be `{head of government}`.

In the context of zero-shot RE, we aim to predict the relation label $\mathbf{Y}$ between the entity pair (x_h, x_t) without explicit training data or demonstration for this specific relation label [34]. Leveraging LLMs is a recent approach that

has garnered attention for addressing zero-shot RE tasks, mainly when a user submits a prompt to an LLM that has not been specifically trained for the RE task described by the prompt [18, 37, 41]. In the domain of zero-shot RE with LLMs, known as generative zero-shot RE, the task involves presenting an input sentence $\mathbf{X}$ alongside a prompt containing task instruction $\mathcal{I}$ and examples. The goal is to generate a relation label $\mathbf{Y} = \{y_1, y_2, \cdots, y_m\}$ consisting of m tokens, representing the relationship between the head (x_h) and tail (x_t) entities mentioned in the input sentence.

It is worth noting that since LLMs are generative, they may generate a relation label $\mathbf{Y}$ consisting of several tokens, whereas, in the other traditional supervised methods excluding generative ones, we mainly consider the relation label as one token. For example, consider the input sentence $\mathbf{X}$ = `{Claude Malhuret is the mayor of Vichy, France.}`. In this case, the task instruction $\mathcal{I}$ might instruct the LLM to identify the relation between x_h = {`Claude Malhuret`} and x_t = {`Vichy`} and generate a relation label. The LLM, after processing the prompt consisting $\mathcal{I}$, $\mathbf{X}$, x_h, and x_t might generate $\mathbf{Y}$ as {`head of government`}, indicating that the head entity is the director of tail entity.

Vanilla Prompting Vanilla prompting represents the most straightforward prompt strategy, involving direct instruction to LLMs to extract relation labels from input sentences. As depicted in Fig. 1 (b), a prompt containing task instructions $\mathcal{I}$, the input sentence $\mathbf{X}$, head entity x_h, head entity x_t, and a pre-defined list of relations $\mathbf{L}$ are provided as input to the LLM. The LLM then generates the relation label $\mathbf{Y}$ (shown in green text).

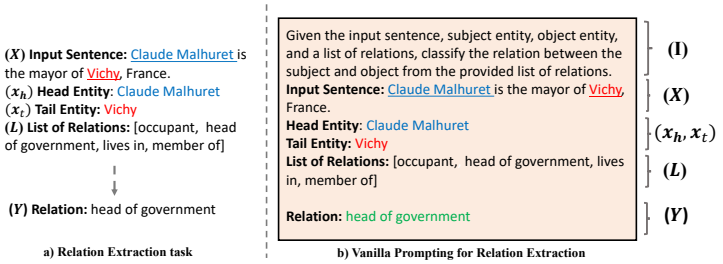

Fig. 1: Illustration of relation extraction task. Panel (a) outlines the task of relation extraction, including the identification of head and tail entities within a given input sentence and the classification of their relation from a predefined list. Panel (b) demonstrates the application of vanilla prompting techniques for relation extraction.

4 Methodology

In this section, we propose the Refine-Estimate-Answer (REA) prompting approach, addressing the complexities of relation extraction (RE) as discussed in Section 1. REA systematically break down the complex task of RE into four distinct stages. Each stage simplifies the process, considering the LLM's refinement of their responses by integrating prior iterations. The overall strategy of REA comprises the following four main stages:

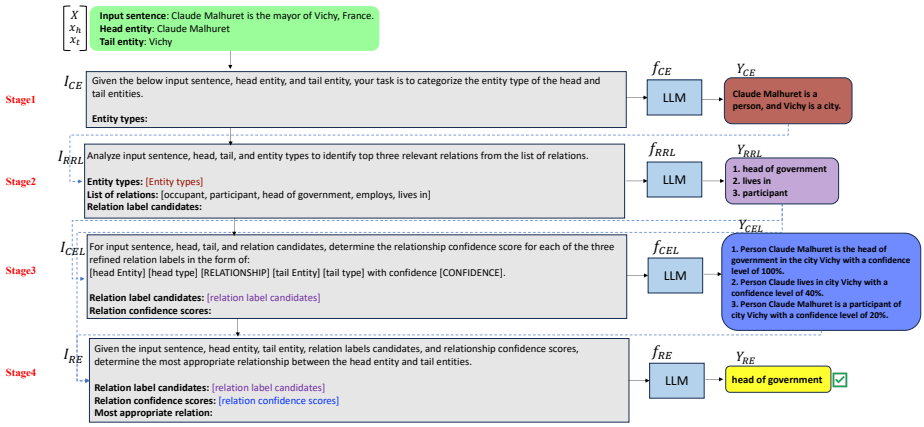

Fig. 2: REA Prompting Overview: The approach comprises four stages, with each stage's response serving as a component within the input to the subsequent stage.

1. *Stage1 (contextual enrichment)*: Given an input sentence $\mathbf{X}$, head entity x_h, and tail entity x_t, LLM determines the types of head and tail entities.
2. *Stage2 (refinement of relation labels)*: LLM refines the list of relation labels based on the extracted entity types from the previous stage.
3. *Stage3 (confidence elicitation)*: LLM generates sentences with head and tail entities for each refined relation label candidate, assigning a confidence percentage to each sentence.
4. *Stage4 (generate final relation label)*: Utilizing the generated sentences and confidence percentages, the LLM determines the predicted relation label, thus finalizing the output.

As shown in Figure 2, each step above involves prompting the LLM in a zero-shot manner to achieve the desired response. The subsequent section provides a detailed description of each stage.

4.1 Stage1: Contextual Enrichment

Studies have demonstrated that enriching RE models with contextual information such as typed entity markers can significantly improve their performance [45]. Considering entity types, knowing whether an entity is a person, an organization, or a location can offer valuable clues about its potential relationship. For instance, encountering a person as the head entity and a city as the tail entity strongly suggests a `"born in"` relationship rather than the less likely `"employs"`. Therefore, in our approach, we leverage contextual enrichment as the first step of our prompting approach to provide the model with additional information about entity types. Specifically, given the input sentence, head entity, tail entity, and a pre-defined list of possible entity types, the LLM is prompted to generate the types of entities. The generated entity types are utilized to refine relation labels. This step can be represented as:

$$\mathbf{Y}_{CE} = f_{\text{CE}}(\mathcal{I}_{CE}, \mathbf{X}, x_h, x_t) \tag{1}$$

The function f_{CE} represents the process of contextual enrichment, where the provided task instruction $\mathcal{I}_{CE}$, input sentence $\mathbf{X}$, head entity x_h, and tail entity x_t are used as prompt to generate the entity types $\mathbf{Y}_{CE}$. As illustrated in Figure 2, with the task instruction $\mathcal{I}_{CE}$ for entity type categorization, considering $\mathbf{X}$ as {`Claude Malhuret is the mayor of Vichy, France.`}, x_h = {`Claude Malhuret`}, and x_t = {`Vichy`}, the resulting $\mathbf{Y}_{CE}$ from the LLM response would be {`Claude Malhuret is a person, and Vichy is a city`}. This process enriches the context with helpful entity information.

4.2 Stage2: Refinement of Relation Labels

Following the generation of entity types through contextual enrichment at Stage1, the subsequent step focuses on refining the relation labels based on these entity types. This refinement process ensures that the selected relation labels align appropriately with the identified entity types. During this stage, the LLM is prompted to choose the three most relevant relations, considering the input sentence, head entity, tail entity, a pre-defined list of relation labels, and generated entity types. The pre-defined list of relation labels encompasses all relation labels present in the dataset. The choice of three candidates for refining relation labels is grounded in empirical experiments, which revealed optimal performance at this level of consideration. This balance ensures both computational efficiency and accurate predictions. This stage can be expressed as:

$$\mathbf{Y}_{RRL} = f_{RRL}(\mathcal{I}_{RRL}, \mathbf{X}, x_h, x_t, \mathbf{Y}_{CE}, \mathbf{L}) \tag{2}$$

The function f_{RRL} refines the relation labels based on the prompt comprised of task instruction $\mathcal{I}_{RRL}$, input sentence $\mathbf{X}$, head entity x_h, tail entity x_t, pre-defined list of labels $\mathbf{L}$, and generated entity types from the previous step $\mathbf{Y}_{CE}$. As depicted in Figure 2, the task instruction $\mathcal{I}_{RRL}$ demonstrating refinement of relation labels $\mathbf{L}$ = {`occupant, participant, head of government, employs, lives in`}, with $\mathbf{X}$ as {`Claude Malhuret is the mayor of Vichy, France.`}, x_h = {`Claude Malhuret`}, x_t = {`Vichy`}, and $\mathbf{Y}_{CE}$ = {`Claude Malhuret is a person, and Vichy is a city`} used to generate relation label candidates $\mathbf{Y}_{RRL}$ = {`1. head of government 2. lives in 3. participant`}.

4.3 Stage3: Confidence Elicitation

The next stage involves confidence elicitation, where the certainty levels associated with the relation label candidates generated by the LLM are estimated. Confidence elicitation involves estimating the certainty levels related to the responses generated by LLMs without relying on accessing specific architectural details or adjusting the model through pre-training or fine-tuning [15]. Tian et al. [31] proposed that LLMs trained with reinforcement learning with human feedback (RLHF), such as GPT-3.5 [25], generally exhibit better-calibrated verbalized confidences emitted as output tokens compared to the model's conditional probabilities. Similarly, our approach integrates verbalized confidence elicitation in our prompting methodology.

In this stage, prompt construction follows a structured format (see Figure 2). Leveraging elements such as head entity, tail entity, and relation label candidates generated in the preceding step, the LLM is prompted to generate sentences in the following form:

```
[head entity type] [head entity] [relation label candidate] [tail entity type] [tail
entity] with a confidence level of [confidence]%.
```

Here, the goal is to determine the confidence level for each relation label candidate. The prompt guides the LLM in replacing the placeholders with actual entities and their respective types alongside each relation candidate. It subsequently generates a confidence score based on the knowledge acquired during the pre-training stage. The confidence elicitation step can be represented as:

$$\mathbf{Y}_{CEL} = f_{CEL}(\mathcal{I}_{CEL}, \mathbf{X}, x_h, x_t, \mathbf{Y}_{RRL}, \mathbf{Y}_{CE}) \tag{3}$$

The function f_{CEL} estimates confidence levels based on the prompt giving instruction $\mathcal{I}_{CEL}$ regarding the task, input sentence $\mathbf{X}$, head entity x_h, tail entity x_t, generated entity types $\mathbf{Y}_{CE}$, and relation label candidates $\mathbf{Y}_{RRL}$. As shown in Figure 2, considering $\mathbf{X}$ as {`Claude Malhuret is the mayor of Vichy, France.`}, x_h = {`Claude Malhuret`}, x_t = {`Vichy`}, $\mathbf{Y}_{CE}$ = {`Claude Malhuret is a person, and Vichy is a city.`}, and relation label candidates $\mathbf{Y}_{RRL}$ = {`1. head of government 2. lives in 3. participant`} the expected generated confidence sentences $\mathbf{Y}_{CEL}$ would be:

1. Person Claude Malhuret is the head of government in the city Vichy with a confidence level of 100%.
2. Person Claude Malhuret lives in city Vichy with a confidence level of 40%.
3. Person Claude Malhuret is a participant of city Vichy with a confidence level of 20%.

4.4 Stage4: Generate Final Relation Label

The ultimate stage determines the most suitable relationship between the provided head and tail entities. This decision relies on the input text $\mathbf{X}$, head entity x_h, tail entity x_t, relation label candidates $\mathbf{Y}_{RRL}$, and the confidence scores associated with each candidate $\mathbf{Y}_{CEL}$:

$$\mathbf{Y}_{RE} = f_{RE}(\mathcal{I}_{RE}, \mathbf{X}, x_h, x_t, \mathbf{Y}_{RRL}, \mathbf{Y}_{CEL}) \tag{4}$$

The function f_{RE} signifies the procedure by which the LLM generates the final relation label $\mathbf{Y}_{RE}$ by processing the task instruction $\mathcal{I}_{RE}$ and the elements mentioned earlier. Figure 2 illustrates that the LLM determined {`head of government`} as the most appropriate label based on the answers provided in previous stages.

5 Experiments

This section outlines our experimental methodology, evaluating the effectiveness of REA prompting in RE tasks under zero-shot scenarios.

5.1 Datasets and Implementation Details

We conducted our experiments using three English RE datasets: FewRel [10], Wiki-ZSL [4], and TACRED [42]. FewRel is a few-shot RE benchmark dataset sourced from Wikipedia, including 80 relations. The Wiki-ZSL dataset comprises 113 relations generated from Wikipedia articles and the Wikidata knowledge base by distant supervision. The TACRED contains 42 relations extracted from news articles. In line with previous studies [4,18], our experiments on the FewRel and Wiki-ZSL datasets involved varying sizes (m) of relation label sets to evaluate method performance. Here, m denotes the number of unique relation labels, with values chosen from $\{5, 10, 15\}$. To ensure robustness against experimental variability, we repeated the label selection process five times using different random seeds, resulting in distinct test sets. Regarding TACRED, in accordance with previous studies [18, 41], to manage OpenAI costs, we randomly selected 1000 examples from the test set. We measured performance with *precision*, *recall*, and *macro-F1* for FewRel and Wiki-ZSL, and *micro-F1* was used for TACRED, excluding the none-of-the-above (NoTA) relation.

For the commercialized LLM, we used *gpt-3.5-turbo* [25] accessed through the OpenAI API. With *gpt-3.5-turbo*, no post-processing is necessary since the model generates a relation label directly. As for the open-source LLM, we chose *Mixtral-8x7B* [12], a pre-trained generative Sparse Mixture of Experts model available on the Hugging Face model hub [1]. For *Mixtral-8x7B*, we extracted the relation label from the model responses, as it sometimes did not provide a relation label alone. All reported scores are averages from five experiments to ensure robustness.

5.2 Baselines

We compare REA against existing zero-shot RE frameworks, dividing them into two groups: (1) traditional supervised methods and (2) generative LLM-based methods. Traditional supervised methods, primarily utilizing PLMs, are designed to leverage labeled RE datasets for training and subsequently generalize to unseen RE datasets, particularly for relation labels. In contrast, generative LLM-based methods use pre-training knowledge to predict relations without fine-tuning on labeled data. The frameworks included in each category are as follows:

- Traditional supervised methods: (1) ESIM [16], a traditional approach using BiLSTM for reading comprehension-based RE; (2) ZS-BERT [4], a supervised method employing BERT for encoding sentences and relation descriptions to classify and predict relations; (3) TGM [19], a generative meta-learning RE framework training T5-base [28] for learning and extracting unseen relations; (4) RelationPrompt [5], a supervised RE framework which uses GPT-2 [27] and BART [17] for generating and extracting relations from

[1] https://huggingface.co/models

Table 1: Main results on FewRel and Wiki-ZSL datasets with $m \in \{5, 10, 15\}$ unique relations. The approaches are divided into traditional supervised and LLM-based models.

Dataset	**FewRel**								
	m=5			m=10			m=15		
	P	R	F1	P	R	F1	P	R	F1
ESIM	56.27	58.44	57.33	42.79	44.17	43.52	29.15	31.59	30.32
ZS-BERT	76.96	78.86	77.90	56.92	57.59	57.25	35.54	38.19	36.82
TGM	39.40	38.91	39.15	30.18	29.77	29.97	25.43	24.94	25.19
RelationPrompt	90.15	88.50	89.30	80.33	79.62	79.96	74.33	72.51	73.40
RE-Matching	90.52	**90.56**	**90.54**	82.12	**81.55**	**81.83**	**73.80**	73.52	**73.66**
Vanilla	67.41	72.97	70.08	42.48	46.26	44.29	25.71	27.77	26.70
SUMASK	78.27	72.55	75.30	64.77	60.94	62.80	44.76	41.13	42.87
REA (Mixtral)	76.19	81.2	78.62	63.18	66.70	64.89	61.23	**80.80**	69.67
REA (GPT-3.5)	**92.57**	84.7	88.46	**82.26**	79.47	80.85	64.34	68.68	66.44
Dataset	**Wiki-ZSL**								
	m=5			m=10			m=15		
	P	R	F1	P	R	F1	P	R	F1
ESIM	48.58	47.74	48.16	44.12	45.46	44.78	27.31	29.62	28.42
ZS-BERT	71.54	72.39	71.96	60.51	60.98	60.74	34.12	34.38	34.25
TGM	40.67	33.42	36.56	26.09	21.84	23.73	22.10	18.27	19.99
RelationPrompt	70.66	**83.75**	76.63	68.51	**74.76**	71.50	63.69	**67.93**	65.74
RE-Matching	78.19	78.41	**78.30**	**74.39**	73.54	**73.96**	**67.31**	67.33	**67.32**
Vanilla	64.47	70.83	67.50	41.83	46.22	43.92	23.17	27.82	25.28
SUMASK	75.64	70.96	73.23	62.31	61.08	61.69	43.55	40.27	41.85
REA (Mixtral)	69.46	50.4	58.41	58.72	53.59	56.04	54.19	48.12	50.97
REA (GPT-3.5)	**78.88**	68.6	73.38	73.15	61.2	66.64	58.2	52.6	55.25

synthetic data; (5) RE-Matching [43], a supervised RE framework, decouples encoding and matching using Sentence-BERT [29] and BERT [33] for relation extraction and feature distillation.

– Generative LLM-based methods: (1) Vanilla, as discussed in Section 3, GPT-3.5 model is prompted to extract relation labels from input sentences directly; (2) QA4RE [41], an LLM-based zero-shot RE framework that converts RE tasks into multiple-choice question-answering, utilizing GPT-3.5 to provide answers; (3) SUMASK [18], a generative zero-shot RE framework that combines GPT-3.5 with a natural language inference module to predict relations.

5.3 Results

Table 1 presents a comparative analysis of zero-shot RE on the FewRel and Wiki-ZSL datasets for both traditional supervised and LLM-based approaches. Notable, the REA prompting approach, specifically when implemented with the GPT-3.5 model (REA (GPT-3.5)), is distinguished by its superior performance among other LLM-based methods across datasets. This achievement is particularly significant when juxtaposed with SUMASK [18]. Despite SUMASK using a decomposition strategy and an external module for uncertainty estimation, REA outperforms it by directly utilizing the core capabilities of LLMs. Furthermore, REA's resilience to the variation in the number of relations (m) contrasts

Table 2: Zero-shot results of LLM-based methods on the TACRED dataset with NoTA relation excluded.

Method	P	R	F1
Vanilla	36.9	68.8	48.1
QA4RE	47.7	**78.6**	59.4
SUMASK	62.2	53.8	57.7
REA (Mixtral)	59.74	56.3	57.97
REA (GPT-3.5)	**72.95**	56.2	**63.49**

with other LLM-based methods, which show a marked decrease in performance with an increase in m. REA, in its implementations with both GPT-3.5 and Mixtral, significantly enhances the Vanilla approach and surpasses other LLM-based frameworks, evidencing the effectiveness of our approach.

Additionally, REA (Mixtral) delivers competitive results in both datasets; particularly notable are the F1 score of 78.62% at $m = 5$ in FewRel and 58.41% at $m = 5$ in Wiki-ZSL. However, REA (GPT-3.5) consistently outperforms these results suggesting the GPT-3.5 model employed in REA (GPT-3.5) offers advantages in zero-shot RE tasks.

An important observation is that while REA may not outperform all traditional supervised models, such as RE-Matching, its ability to operate effectively without the need for fine-tuning on relation-specific data stands out as a significant advantage over traditional supervised approaches. These traditional methods typically depend on training with annotated data to predict unseen relations. The inherent flexibility of REA, originating from its independence from fine-tuning, substantially reduces the time and resources needed for model deployment and adaptation. This benefit makes REA a desirable solution for real-world applications where rapid or broad-scale implementation is necessary. Through REA, we showcase the LLM's capabilities in delivering competitive or superior performance without the extensive training process, highlighting the impact of strategic prompting in enhancing zero-shot RE performance.

Table 2 presents the zero-shot performance of LLM-based methods on the TACRED dataset, explicitly excluding the NoTA (none of the above) relation. This comparative evaluation highlights REA's standout performance. Despite using the same selection of 1000 randomly chosen test records to ensure a fair comparison, REA outperforms other LLM-based models. Notably, REA achieves this performance without relying on ground truth entity types used in QA4RE and SUMASK. This highlights REA's effective use of the contextual enrichment phase (see Section 4.1), which can intuitively capture the nuances of entity types from text alone. Beyond outperforming other LLM-based methods, REA demonstrates consistency and effectiveness across different LLM architectures, including GPT-3.5 and Mixtral, showcasing its ability to improve upon the vanilla prompting approach. Moreover, the notably high precision achieved by REA, particularly with the GPT-3.5, signifies its ability to accurately identify relevant relations without generating excessive false positives, a critical attribute supporting high-quality relation extraction.

5.4 Analysis

In this section, we analyze the REA approach, focusing on its effectiveness and the role of its components in zero-shot RE. Our analysis evaluates the impact of consolidating REA's stages into fewer steps and investigates the contributions of critical stages such as contextual enrichment and confidence elicitation. This exploration aims to highlight REA's strengths, areas for improvement and insights into developing effective prompting strategies for LLMs in zero-shot RE tasks.

Consolidated Steps Strategy Analysis: Our investigation explored the effectiveness of integrating specific REA phases—contextual enrichment with relation label refinement and confidence elicitation with final relation label generation—into unified steps. Our Objective was to assess how such a unified approach might affect overall model performance, utilizing a singular prompt to integrate stages. Table 3 compares the performance of the consolidated REA approach, which combines stages 1 and 2, as well as stages 3 and 4, with the original REA method that maintains distinct stages across various datasets.

The analysis reveals a prominent trend across the FewRel, Wiki-ZSL, and TACRED datasets. The consolidated REA method, which merges the initial and final stages of the process, consistently underperforms compared to the original, stepwise REA approach. Specifically, in the FewRel dataset, the F1 scores for the consolidated method range from 45.26 to 75.47 across different relation counts ($m = 5, 10, 15$), whereas the original REA approach maintains higher scores, peaking at 88.46. This pattern is mirrored in the Wiki-ZSL and TACRED datasets, where the original REA method surpasses the consolidated version, achieving F1 scores up to 73.38 and 63.49, respectively. This uniform drop in performance with the consolidated approach suggests that merging steps might obscure crucial intermediate information or feedback loops inherent to the original REA methodology. Each discrete stage in the original REA potentially offers a unique opportunity for refinement and calibration based on specific aspects of the relation extraction task. By collapsing these stages, the consolidated method likely loses the chance to iteratively adjust its strategy based on feedback from each phase, thus limiting its ability to accurately capture and reflect the complexities of the RE task. This insight underscores the value of discrete, focused steps in the REA method, enabling a more effective arrangement with the challenges of zero-shot RE.

Table 3: Comparison of F1 scores between Consolidated REA and Original REA approaches across different datasets.

Dataset	**FewRel**			**Wiki-ZSL**			**TACRED**
	m=5	m=10	m=15	m=5	m=10	m=15	NoTA
Consolidated REA (GPT-3.5)	75.47	64.93	45.26	66.63	54.59	49.01	54.23
Original REA (GPT-3.5)	**88.46**	**80.85**	**66.44**	**73.38**	**66.64**	**55.25**	**63.49**

Evaluating the Impact of Individual Stages in REA: In the comparison depicted in Figure 3, we investigated the individual contributions of contextual enrichment and confidence elicitation stages to the REA prompting approach. We evaluated their respective impacts on the model's overall performance by selectively omitting these stages.

Including contextual enrichment consistently led to higher F1 scores across the datasets, underscoring its pivotal role in aiding the LLM's relation extraction capabilities. Conversely, the absence of this stage resulted in a notable decrease in performance, demonstrating the stage's vital role in enhancing the model's contextual understanding and ability to determine relations within the text.

The necessity of confidence elicitation, however, varied with the complexity of the task. Particularly in the Wiki-ZSL dataset with $m = 5$, the model performed better without this stage, which may suggest that for simpler tasks with fewer relations, the model can effectively deduce the most likely relations without needing to evaluate its own confidence. Nevertheless, the significant decline in F1 scores when confidence elicitation was removed in both datasets reveals its importance. It contributes to accuracy by allowing the model to assess its own certainty, refining its predictions.

These insights confirm that each stage in the REA method is critical in realizing high-performing zero-shot RE. Contextual enrichment lays the foundation for accurate relation understanding, while confidence elicitation tunes the output, contributing to an effective model for zero-shot RE.

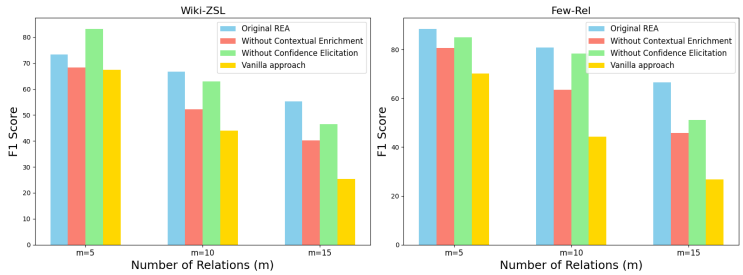

Fig. 3: Comparison of F1 scores across different approaches using GPT-3.5 on Wiki-ZSL and Few-Rel datasets with varying numbers of relations (m=5, m=10, m=15).

6 Conclusion

This paper has introduced the Refine-Estimate-Answer (REA) prompting approach that significantly enhances zero-shot RE by structurally decomposing the task into manageable steps and using iterative refinement to improve performance. Demonstrating superior performance across benchmarks, REA showcases the effectiveness of multi-stage prompting and self-refinement in leveraging LLMs for complex NLP tasks. The success of REA across various LLMs, including open-source and commercialized models, vows its versatility and potential for broader NLP tasks. Future work will explore optimizing these mechanisms and extending REA's methodology to other information extraction tasks facing

data scarcity. This study advances zero-shot RE and sets a new standard for employing LLMs in NLP, highlighting the importance of structured prompting strategies for maximal model performance.

References

1. Arora, S., Narayan, A., Chen, M.F., et al.: Ask me anything: A simple strategy for prompting language models. In: ICLR (2023), https://openreview.net/forum?id=bhUPJnS2g0X
2. Brown, T., Mann, B., Ryder, N., et al.: Language models are few-shot learners. NeurIPS **33**, 1877–1901 (2020)
3. Bunescu, R., Mooney, R.: A shortest path dependency kernel for relation extraction. In: EMNLP. pp. 724–731 (2005)
4. Chen, C.Y., Li, C.T.: Zs-bert: Towards zero-shot relation extraction with attribute representation learning. In: NAACL. pp. 3470–3479 (2021)
5. Chia, Y.K., et al.: Relationprompt: Leveraging prompts to generate synthetic data for zero-shot relation triplet extraction. In: Findings of the ACL 2022
6. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555 (2014)
7. Deng, S., Ma, Y., Zhang, N., Cao, Y., Hooi, B.: Information extraction in low-resource scenarios: Survey and perspective. arXiv preprint arXiv:2202.08063 (2022)
8. Dhuliawala, S., Komeili, M., et al.: Chain-of-verification reduces hallucination in large language models. arXiv preprint arXiv:2309.11495 (2023)
9. Ding, N., Wang, X., Fu, Y., et al.: Prototypical representation learning for relation extraction. In: ICLR 2021
10. Han, X., Zhu, H., et al.: Fewrel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In: EMNLP. pp. 4803–4809 (2018)
11. Huang, F., Kwak, H., An, J.: Chain of explanation: New prompting method to generate quality natural language explanation for implicit hate speech. In: The ACM Web Conference 2023
12. Jiang, A.Q., Sablayrolles, A., et al.: Mixtral of experts. arXiv preprint arXiv:2401.04088 (2024)
13. Khoo, C., Myaeng, S.H.: Identifying semantic relations in text for information retrieval and information extraction. In: The semantics of relationships: An interdisciplinary perspective, pp. 161–180 (2002)
14. Kojima, T., Gu, S.S., et al.: Large language models are zero-shot reasoners. NIPS **35**, 22199–22213 (2022)
15. Kuhn, L., Gal, Y., Farquhar, S.: Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In: ICLR (2023)
16. Levy, O., Seo, M., et al.: Zero-shot relation extraction via reading comprehension. In: CoNLL. pp. 333–342 (2017)
17. Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., et al.: BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: ACL. pp. 7871–7880 (Jul 2020)
18. Li, G., Wang, P., Ke, W.: Revisiting large language models as zero-shot relation extractors. In: Findings of EMNLP 2023. pp. 6877–6892
19. Li, W., Qian, T.: Generative meta-learning for zero-shot relation triplet extraction. arXiv preprint arXiv:2305.01920 (2023)
20. Liu, J., Shen, D., pthers: What makes good in-context examples for gpt-3? In: DeeLIO 2022. pp. 100–114 (2022)

21. Liu, P., et al.: Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. ACM Computing Surveys (2023)
22. Luo, D., Su, J., Yu, S.: A bert-based approach with relation-aware attention for knowledge base question answering. In: 2020 IJCNN. IEEE (2020)
23. Madaan, A., et al.: Self-refine: Iterative refinement with self-feedback. NIPS **36** (2024)
24. Muhammad, I., Kearney, A., et al.: Open information extraction for knowledge graph construction. In: DEXA. pp. 103–113 (2020)
25. OpenAI: Introduce chatgpt. OpenAI blog (2023), https://openai.com/blog/chatgpt
26. Press, O., Zhang, M., et al.: Measuring and narrowing the compositionality gap in language models. In: Findings of EMNLP. pp. 5687–5711 (2023)
27. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al.: Language models are unsupervised multitask learners. OpenAI blog **1**(8), 9 (2019)
28. Raffel, C., Shazeer, N., et al.: Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of Machine Learning Research **21**, 1–67 (2020)
29. Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. In: EMNLP (2019)
30. Shi, F., Suzgun, M., et al.: Language models are multilingual chain-of-thought reasoners. arXiv preprint arXiv:2210.03057 (2022)
31. Tian, K., Mitchell, E., Zhou, A., et al.: Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In: EMNLP. pp. 5433–5442 (2023)
32. Tourille, J., Ferret, O., Neveol, A., et al.: Neural architecture for temporal relation extraction: A bi-lstm approach for detecting narrative containers. In: ACL (2017)
33. Vaswani, A., Shazeer, N., et al.: Attention is all you need. Advances in neural information processing systems (2017)
34. Wang, W., Zheng, V.W., et al.: A survey of zero-shot learning: Settings, methods, and applications. ACM TIST pp. 1–37 (2019)
35. Wang, Z., Wen, R., et al.: Finding influential instances for distantly supervised relation extraction. In: COLING. pp. 2639–2650 (2022)
36. Wei, J., Wang, X., et al.: Chain-of-thought prompting elicits reasoning in large language models. NIPS **35**, 24824–24837 (2022)
37. Wei, X., Cui, X., et al.: Zero-shot information extraction via chatting with chatgpt. arXiv preprint arXiv:2302.10205 (2023)
38. Xu, D., Chen, W.: Large language models for generative information extraction: A survey. arXiv preprint arXiv:2312.17617 (2023)
39. Yasunaga, M., Chen, X., Li, Y., Pasupat, P., Leskovec, J., et al.: Large language models as analogical reasoners. arXiv preprint arXiv:2310.01714 (2023)
40. Yu, W., Zhang, H., et al.: Chain-of-note: Enhancing robustness in retrieval-augmented language models. arXiv preprint arXiv:2311.09210 (2023)
41. Zhang, K., Jimenez Gutierrez, B.: Aligning instruction tasks unlocks large language models as zero-shot relation extractors. In: Findings of ACL 2023
42. Zhang, Y., Zhong, V., et al.: Position-aware attention and supervised data improve slot filling. In: Proceedings of EMNLP Conference (2017)
43. Zhao, J., Zhan, W., Zhao, W.X., et al.: Re-matching: A fine-grained semantic matching method for zero-shot relation extraction. In: ACL. pp. 6680–6691 (2023)
44. Zhou, D., Schärli, N., et al.: Least-to-most prompting enables complex reasoning in large language models. In: ICLR (2022)
45. Zhou, W., Chen, M.: An improved baseline for sentence-level relation extraction. In: ACL (Short Papers). pp. 161–168 (2022)