When Phrases Meet Probabilities: Enabling Open Relation Extraction with Cooperating Large Language Models

Jiaxin Wang[1,3] **, Lingling Zhang**[1,3*] **, Wee Sun Lee**[2] **, Yujie Zhong**[1] **, Liwei Kang**[2]**, Jun Liu**[1,4]
[1]School of Computer Science and Technology, Xi'an Jiaotong University
[2] School of Computing, National University of Singapore
[3] Ministry of Education Key Laboratory of Intelligent Networks and Network Security, China
[4] Shaanxi Province Key Laboratory of Big Data Knowledge Engineering, China
jiaxinwangg@outlook.com {zhanglling,liukeen}@xjtu.edu.cn
{leews,kang}@comp.nus.edu.sg yjzhong6@gmail.com

Abstract

Current clustering-based open relation extraction (OpenRE) methods usually apply clustering algorithms on top of pre-trained language models. However, this practice has three drawbacks. First, embeddings from language models are high-dimensional and anisotropic, so using simple metrics to calculate distances between these embeddings may not accurately reflect the relational similarity. Second, there exists a gap between the pre-trained language models and downstream clustering for their different objective forms. Third, clustering with embeddings deviates from the primary aim of relation extraction, as it does not directly obtain relations. In this work, we propose a new idea for OpenRE in the era of LLMs, that is, extracting relational phrases and directly exploiting the knowledge in LLMs to assess the semantic similarity between phrases without relying on any additional metrics. Based on this idea, we developed a framework, ORELLM, that makes two LLMs work collaboratively to achieve clustering and address the above issues. Experimental results on different datasets show that ORELLM outperforms current baselines by $1.4\% \sim 3.13\%$ in terms of clustering accuracy.

1 Introduction

Relation Extraction (RE) focuses on extracting relations between entity pairs from texts. However, conventional RE is based on the closed-set hypothesis (Liu et al., 2022; Yan et al., 2023; Xu et al., 2023) for handling pre-defined relations, thereby limiting its capability to deal with new emerging relations in the real world. Therefore, Open Relation Extraction (OpenRE), which aims to extract new relations from unlabeled open-domain corpora, has received more attention.

One paradigm of OpenRE is open information extraction (OpenIE) (Etzioni et al., 2008; Fader et al., 2011), which directly extracts relation-related phrases from sentences based on syntactic analysis, but often produces redundant or ambiguous results. Therefore, another paradigm, namely unsupervised relation discovery (Yao et al., 2011; ElSahar et al., 2017), has been proposed to bridge the gap between closed-set RE and OpenIE by inducing new relations automatically. Specifically, it commonly formulates OpenRE as a clustering task, where the most critical challenge lies in how to measure the relational similarity between unlabeled instances effectively for clustering. For this purpose, a popular practice (as shown in Figure 1(a)) is to first obtain embeddings from pre-trained language models such as BERT (Devlin et al., 2019), and then apply clustering algorithms, e.g., K-means with euclidean, to measure the distances of these embeddings and form clusters (Hu et al., 2020; Tran et al., 2020). Although this practice benefits from the knowledge in pre-trained language models, it has three drawbacks. **First**, the embeddings from pre-trained language models encode complex linguistic information (Reif et al., 2019; Zhao et al., 2021) and are anisotropic (Gao et al., 2019; Li et al., 2020). Consequently, using simple metrics, e.g.,

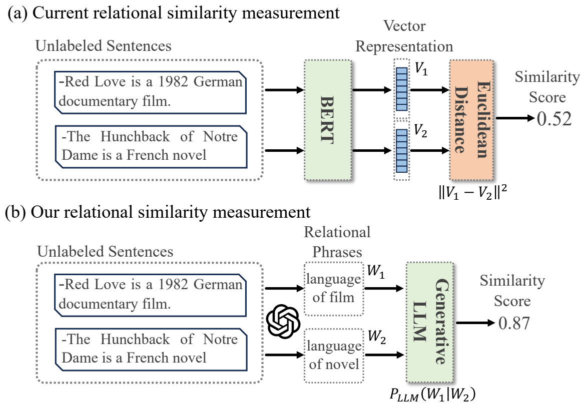


Figure 1: Comparison of relational similarity measurement methods.

* Corresponding author

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 13130–13147
August 11-16, 2024 ©2024 Association for Computational Linguistics

Euclidean or cosine, to calculate distances between these embeddings may not accurately reflect the relational similarity. **Second**, this practice exists a gap between the pre-trained language models and downstream clustering for their different objective forms (Wang et al., 2020; Qiu et al., 2020). In other words, the pre-training tasks of language models (i.e., masked language modeling or next token prediction) are not consistent with the downstream clustering task (i.e., classification with pseudo labels), which restricts the transfer and adaptation of knowledge in pre-trained language models to unsupervised clustering. **Third**, the essence of these methods is to perform clustering with embeddings, which does not meet the original intention of RE since the relations in each sentence are not actually extracted. While some methods (Zhao et al., 2021; Wang et al., 2022; Li et al., 2022) alleviate the first issue by using labeled instances with pre-defined relations to guide similarity learning in new types, they still fail to address the other two drawbacks.

Recently, a series of powerful generative large language models (LLMs) have been proposed and brought revolutionary changes to the field of NLP (Brown et al., 2020; Touvron et al., 2023; OpenAI, 2023). We believe that LLMs have the potential to solve the above problems for two reasons. For one thing, LLMs are trained on a large amount of general text corpora, and have excellent generalization in understanding and generating text (Song et al., 2023). This enables them to analyze and produce relational phrases for unlabeled instances. For another thing, their generation strategy follows "next token prediction" (Radford et al., 2019), which predicts the next token by calculating the likelihood probability of this token conditioned on its preceding sequence. This probability indicates the relatedness degree of the predicted token to its entire prefix and provides a new way to measure the similarity between phrases.

In this work, we propose a novel framework, namely ORELLM, for OpenRE. As illustrated in Figure 1(b), the key of ORELLM is to adopt LLMs in two aspects: generating relational phrases for relation extraction and calculating conditional probabilities between phrases for similarity learning. Specifically, ORELLM consists of an API-based LLM and an open-source LLM. The former serves as a phrase extractor, generating relational phrases from instances. The latter operates as a probability estimator, computing the likelihood that one phrase generates another, thereby reflecting the semantic similarity among the generated phrases. As a result, the probability estimator directly exploits the knowledge in LLMs to assess the semantic relatedness between phrases, without relying on any additional metrics. Moreover, we propose a cooperative cycle to make these two LLMs work collaboratively for clustering. In this cycle, we first select the initial centers and assign each data point to the corresponding centers according to probabilities. We then sort the data in each cluster by the magnitude of probability scores to obtain reliable and ambiguous data. After that, we design a "Rethink" strategy to correct the potentially erroneous phrases in ambiguous data and discover new centers. In addition, we adopt the reliable data to fine-tune the probability estimator. This process is consistent with the next token prediction pattern, thus alleviating the gap between pre-trained language models and the downstream clustering task. Finally, we use the fine-tuned probability estimator to reassign labels and update clustering for the next cycle.

To summarize, the main contributions of our work are as follows: (1) We propose a new idea for OpenRE in the era of LLMs, that is, first extract relational phrases and then directly exploit the knowledge in LLMs to assess the semantic similarity between phrases, without relying on any additional metrics. (2) Based on this idea, we propose a framework ORELLM that makes two LLMs work collaboratively for clustering. (3) We conduct experiments on two datasets and ORELLM achieves better performance than the existing state-of-the-art methods. (4) Additionally, ORELLM can output relational phrases for each instance, which really meets the original intention of RE.

2 Methodology

2.1 Problem Formulation

Let $\mathcal{D} = \langle s_i, h_i, t_i \rangle_{i=1}^{N}$ denote the open relational corpus with N instances. Each instance includes a sentence s_i along with its corresponding head entity h_i and tail entity t_i. OpenRE aims to discover relations in $\mathcal{D}$ without annotation labels or pre-defined types. Therefore, it is commonly formulated as a clustering task that clusters instances in $\mathcal{D}$ to K groups and outputs the clustering assignment $\mathcal{Y} = \{y_i\}_{i=1}^{N}$ for instances.

2.2 Overview

As depicted in Figure 2, our ORELLM consists of two LLMs working in three main procedures: (1)

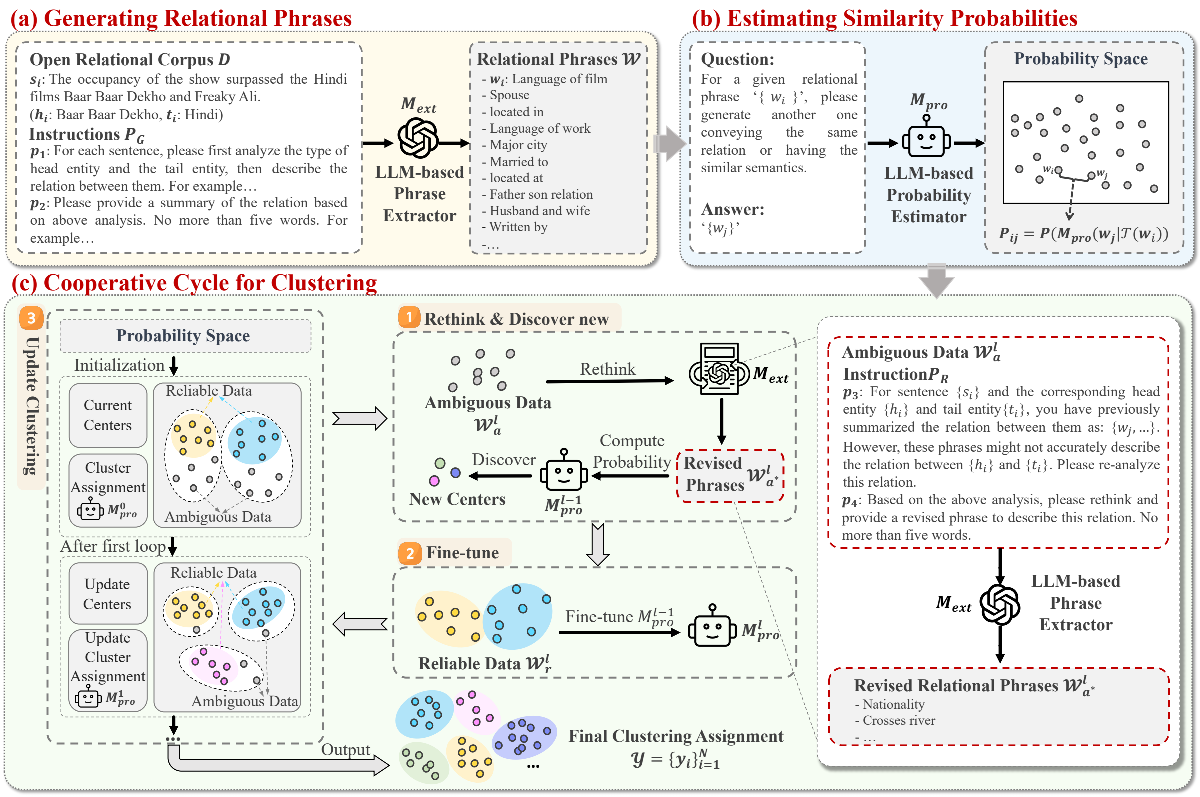


Figure 2: Illustration of our ORELLM framework. In the cooperative cycle, (1) the ambiguous data serves as feedback to prompt $\mathcal{M}_{ext}$ to rethink. Then, we check for any new centers and add them to the existing center set; (2) We use reliable data to fine-tune $\mathcal{M}_{pro}$; (3) We update centers and reassign labels for the next cycle.

Utilizing an API-based LLM as a phrase extractor (denoted as $\mathcal{M}_{ext}$) to generate relational phrases for instances in $\mathcal{D}$; (2) Employing an open-source LLM as a probability estimator (denoted as $\mathcal{M}_{pro}$) to compute the likelihood of one phrase generating another. This procedure outputs a probability space that reflects the semantic similarity among the generated phrases. (3) After cluster initialization, we filter phrases based on current clustering results, obtaining reliable and ambiguous data. Subsequently, we establish a cooperative cycle with three steps to make $\mathcal{M}_{ext}$ and $\mathcal{M}_{pro}$ work collaboratively for cluster learning. First, the ambiguous data serves as feedback to prompt $\mathcal{M}_{ext}$ to rethink, i.e., revise current phrases to generate new ones. We then check these revised phrases for any new centers and add them to the existing center set. Second, we use reliable data to fine-tune $\mathcal{M}_{pro}$. Third, we update centers and reassign labels for the next cycle. The details are as follows.

2.3 Generating Relational Phrases

Previous research demonstrates that prompting LLMs to generate intermediate rationales can greatly improve their performance on specific tasks (Wadhwa et al., 2023; Wan et al., 2023). Therefore, as shown in Figure 2(a), we construct instructions $\mathcal{P_G} = \{p_1, p_2\}$ for each input sentence s_i. Here, p_1 guides $\mathcal{M}_{ext}$ to identify the roles of h_i and t_i and to analyze the semantic associations between them, while p_2 instructs $\mathcal{M}_{ext}$ to summarize the corresponding relational phrases based on this analysis. We also provide a demonstration in $\mathcal{P_G}$ to make $\mathcal{M}_{ext}$ better understand and execute instructions[1]. Note that the relation of this demo is different from all relations in $\mathcal{D}$. In short, this procedure can be formulated as:

$$\boldsymbol{w}_i = \mathcal{M}_{ext}(\mathcal{P_G}, \langle s_i, h_i, t_i\rangle | \langle s_i, h_i, t_i\rangle \in \mathcal{D}), \quad (1)$$

where $\boldsymbol{w}_i \in \mathcal{W}$ represents the generated relational phrase for the i-th input instance.

2.4 Estimating Similarity Probabilities

One characteristic of relations is that they have a diversity of expressions, which means that even the same relation can be expressed differently depending on different contexts. Consequently, the obtained relational phrases are also diverse as they

[1] More details are in Appendix A.4.

are summarized from various instances. For further clustering, we need to quantify the similarity of these phrases efficiently. As illustrated in Figure 2(b), we adopt an open-source generative language model $\mathcal{M}_{pro}$ as a similarity estimator. Specifically, we leverage $\mathcal{M}_{pro}$ to compute the likelihood of $\boldsymbol{w}_j$ conditioned on $\boldsymbol{w}_i$ as the semantic similarity score of $\boldsymbol{w}_i$ and $\boldsymbol{w}_j$. To facilitate this, we design a template $\mathcal{T}$ that transforms $\boldsymbol{w}_i$ into $\mathcal{T}(\boldsymbol{w}_i)$ as: "*Q: For a given relational phrase <$\boldsymbol{w}_i$>, please generate another one conveying the same relation or having similar semantics. A: <?>.*" Then, we feed $\mathcal{T}(\boldsymbol{w}_i)$ into an open-source LLM $\mathcal{M}_{pro}$ and calculate the average conditional probability for generating $\boldsymbol{w}_j$ in the log domain (denoted by $\hat{P}_{ij}$) as follows:

$$\begin{aligned}\hat{P}_{ij} &= \log P_{\mathcal{M}_{pro}}(\boldsymbol{w}_j|\mathcal{T}(\boldsymbol{w}_i)) \\ &= \frac{1}{Q_j}\sum_{q=1}^{Q_j}\log P(w_j^q|\mathcal{T}(\boldsymbol{w}_i), w_j^{<q}),\end{aligned} \tag{2}$$

where Q_j is the token number of $\boldsymbol{w}_j$ and w_j^q represents q-th token in $\boldsymbol{w}_j$. We also compute the probability of $\boldsymbol{w}_i$ conditioned on $\boldsymbol{w}_j$ based on this formula, and the final semantic similarity score between $\boldsymbol{w}_i$ and $\boldsymbol{w}_j$ is $P_{ij} = (\hat{P}_{ij} + \hat{P}_{ji})/2$. In this way, we derive a matrix P from probability space that reflects similarity scores between phrases.

2.5 Cooperative Cycle for Clustering

After extracting relational phrases and estimating similarity, we design a cooperative cycle that enables $\mathcal{M}_{ext}$ and $\mathcal{M}_{pro}$ work collaboratively for clustering. As shown in Figure 2(c), this cycle includes three steps: (i) Rethink and discover new; (ii) Fine-tune; (iii) Update clustering. In each cycle, $\mathcal{M}_{ext}$ is responsible for revising those potentially incorrect phrases, and $\mathcal{M}_{pro}$ takes charge of updating similarity scores. The details are as follows.

Initialization. Before we start the cooperative cycle, we first select initial centers. The similarity score matrix P reflects the distribution of instances in probability space, i.e., higher scores correspond to closer positions, and vice versa. As a result, phrases expressing the same relation tend to be dense, while those conveying different relations tend to be dispersed in this space. On this basis, the initial cluster centers are selected by following two guidelines (Zhao et al., 2023): (1) they have a high density around them, and (2) they are separated from each other. Specifically, inspired by the density-based HDBSCAN clustering algorithm (McInnes et al., 2017), we define the density of $\boldsymbol{w}_i$ as the number of neighbors within a specified radius P_ϵ by:

$$\rho(\boldsymbol{w}_i) = \sum_{j=1}^{N}\mathbb{I}(P_{ij} - P_\varepsilon) > 0, \tag{3}$$

where $\rho(\cdot)$ and $\mathbb{I}(\cdot)$ are the density calculation function and indicator function, respectively. After that, we set a density threshold N_ε and obtain a high-density phrase set, denoted as $\mathcal{W}_{\text{High}} = \{\boldsymbol{w}_i|\rho(\boldsymbol{w}_i) > N_\varepsilon\}$. Then we sort $\boldsymbol{w}_i \in \mathcal{W}_{\text{High}}$ in descending order by the value of $\rho(\boldsymbol{w}_i)$ and take the phrase with the highest density as the first center. Subsequent phrases are added as centers only if their similarity score between any existing centers is lower than the center similarity threshold P_r. From the above process, we can obtain an initial center set $\mathcal{W}^0_{\text{Center}}$, where for $\forall \boldsymbol{w}_i, \boldsymbol{w}_j \in \mathcal{W}^0_{\text{Center}}$, $P_{ij} < P_r$, $\rho(\boldsymbol{w}_i) > N_\varepsilon$, and $\rho(\boldsymbol{w}_j) > N_\varepsilon$. Finally, we assign each remaining phrase to the center with the highest similarity score and obtain the initial clustering assignment result $\mathcal{Y}^0$.

However, the clustering assignment results $\mathcal{Y}^{\ell-1}$ obtained from cycle $\ell - 1$ always have noise because they are actually pseudo-labels. To facilitate subsequent clustering learning, we filter the data based on $\mathcal{Y}^{\ell-1}$, obtaining reliable and ambiguous data for the next cycle. Intuitively, the higher similarity score between the phrase and its center, the more accurate label of this phrase. Therefore, we sort the similarity scores within each cluster in descending order and keep the top $\gamma\%$ of phrases to form the reliable dataset $\mathcal{W}^\ell_r$, while taking the rest as the ambiguous data $\mathcal{W}^\ell_a$.

Rethink and Discover New. We consider that ambiguous data $\mathcal{W}^\ell_a$ exists for two reasons. One is that the current relational phrases generated by $\mathcal{M}_{ext}$ are incorrect. The other is that these phrases do not correspond to any existing centers. Therefore, as shown in Figure 2(c), we design a "Rethink" strategy to guide $\mathcal{M}_{ext}$ in revising previous phrases to obtain a revised phrase set $\mathcal{W}^\ell_{a*}$. Specifically, we devise instructions $\mathcal{P}_\mathcal{R} = \{p_3, p_4\}$ for $w_i \in \mathcal{W}^\ell_a$. In p_3, previously generated phrases serve as feedback to guide $\mathcal{M}_{ext}$ in reanalyzing the relation between head and tail entities. p_4 instructs $\mathcal{M}_{ext}$ to rethink and generate the revised phrases. Then, we employ $\mathcal{M}^{\ell-1}_{pro}$ to calculate the similarity score on $\mathcal{W}^\ell_{a*}$. After that, we use the same selection method as described in the initialization phase to discover new centers $\mathcal{W}^{\ell}_{\text{New}}$ from these revised phrases.

Fine-tune. We construct question-answer pairs from $\mathcal{W}^{\ell}_{r}$ for each cluster, which are denoted as $\mathcal{Z}^{\ell} = \{(\mathcal{T}(\boldsymbol{w}_i), \boldsymbol{w}_j) \mid y_i = y_j, \boldsymbol{w}_i \in \mathcal{W}^{\ell}_{\text{Center}}, \boldsymbol{w}_j \in \mathcal{W}^{\ell}_{r}\}$. Next, $\mathcal{M}^{\ell-1}_{pro}$ is fine-tuned on $\mathcal{Z}^{\ell}$ to produce high generation probabilities for phrases in same clusters by:

$$\mathcal{L} = \mathbb{E}_{\mathcal{Z}^{\ell}} \log P_{\mathcal{M}^{\ell-1}_{pro}}(\mathcal{T}(\boldsymbol{w}_i), \boldsymbol{w}_j), \tag{4}$$

where $(\mathcal{T}(\boldsymbol{w}_i), \boldsymbol{w}_j) \in \mathcal{Z}^{\ell}$. After fine-tuning, we obtain the updated probability estimator $\mathcal{M}^{\ell}_{pro}$, which will be used for further label reassignment.

Update Clustering. In this step, we add the newly discovered center $\mathcal{W}^{\ell}_{\text{New}}$ to the center set $\mathcal{W}^{\ell-1}_{\text{Center}}$ in the last cycle to form the updated center set $\mathcal{W}^{\ell}_{\text{Center}} = \mathcal{W}^{\ell}_{\text{New}} \cup \mathcal{W}^{\ell-1}_{\text{Center}}$ for this cycle. Then, we use $\mathcal{M}^{\ell}_{pro}$ to calculate the similarity scores of all phrases between centers and reassign labels, thereby obtaining the clustering assignment result $\mathcal{Y}^{\ell}$ in cycle ℓ. Based on $\mathcal{Y}^{\ell}$, we filter the data to separate reliable and ambiguous data, preparing for the next cycle of learning. The learning process of ORELLM is summarized in Algorithm 1. Depending on whether K is known, we implement two different stopping conditions. First, in the case of unknown K, we stop after two consecutive cycles without discovering new relations. Second, in the case of known K, we calculate the estimated error rate of K in cycle ℓ as:

$$\text{Err}(K)^{\ell} = |\mathcal{W}^{\ell}_{\text{Center}} - K|/K. \tag{5}$$

The cycle will stop when $\text{Err}(K)^{\ell} > \text{Err}(K)^{\ell-1}$.

Algorithm 1: ORELLM

Input: Open dataset $\mathcal{D} = \{\langle s_i, h_i, t_i \rangle\}_{i=1}^{N}$, Phrase extractor $\mathcal{M}_{ext}$, Probability estimator $\mathcal{M}_{pro}$

Set the cycle $\ell \leftarrow 1$ and initialize $\mathcal{M}^{0}_{pro}$ with pre-trained weights;

Get relational phrases $\mathcal{W} = \{\boldsymbol{w}_i\}_{i=1}^{N}$ for instance in $\mathcal{D}$ by $\mathcal{M}_{ext}$;

Get similarity matrix P in probability space by $\mathcal{M}^{0}_{pro}$;

Select to obtain initial center set $\mathcal{W}^{0}_{\text{Center}}$;

while $\ell = 1$ ***or*** *the conditions for stopping are not met* **do**

  Filter to get reliable data set $\mathcal{W}^{\ell}_{r}$ and ambiguous data set $\mathcal{W}^{\ell}_{a}$;

  `// Rethink and Discover new`

  Do rethink $\mathcal{W}^{\ell}_{a*} \leftarrow \mathcal{W}^{\ell}_{a}$;

  Compute similarity between $\mathcal{W}^{\ell}_{a*}$ by $\mathcal{M}^{\ell-1}_{pro}$, discover new centers set $\mathcal{W}^{\ell}_{\text{New}}$;

  `// Fine-tune`

  Construct question-answer pair set $\mathcal{Z}^{\ell}$;

  $\mathcal{M}^{\ell}_{pro} \leftarrow$ Fine-tune $\mathcal{M}^{\ell-1}_{pro}$ with $\mathcal{Z}^{\ell}$ by Eq. (4);

  `// Update clustering`

  Update center set by $\mathcal{W}^{\ell}_{\text{Center}} \leftarrow \mathcal{W}^{\ell}_{\text{New}} \cup \mathcal{W}^{\ell-1}_{\text{Center}}$;

  Obtain updated clustering assignment $\mathcal{Y}^{\ell}$ by $\mathcal{M}^{\ell}_{pro}$;

  $\ell \leftarrow \ell + 1$;

end

Output: Clustering assignment $\mathcal{Y}^{\ell}$

3 Experiments

3.1 Experimental Settings

Datasets. We evaluate our ORELLM on two widely used relation extraction datasets: FewRel (Han et al., 2018) and TACRED (Zhang et al., 2017). FewRel contains 80 relations, each with 700 instances. Following (Zhang et al., 2023b), we treat the 64 relation types in the original training set as new relations and construct the open relational corpus by randomly selecting 70 instances from each type. For TACRED, we filter out the instances under the "no_relation" label and remove the relation types with less than 70 instances. Afterward, we randomly select 70 instances of each class from the remaining 27 relation types to construct the open relational corpus.

Models. We adopt ChatGPT [2](`gpt-3.5-turbo`) or GPT-4 [3](`gpt-4`) as our phrase extractor. For probability estimator, we utilize a publicly available GPT2-XL model (Radford et al., 2019) with 1.5B parameters. For more details, please refer to Appendix A.1.

Baselines. To demonstrate the effectiveness of our method, we compare it with the following six OpenRE baselines, which are categorized into two groups: (1) Three traditional methods with pre-defined labeled instances for pre-training: **RoCORE** (Zhao et al., 2021), **MatchPrompt** (Wang et al., 2022), and **TABS** (Li et al., 2022). (2) Three methods using LLMs: **ChatGPT** **+RoBERTa-large**, **SBERT**$_{\textbf{ChatGPT +RoBERTa-large}}$,

[2]https://chat.openai.com/chat.

[3]https://chatgpt.com/?oai-dm=1&model=gpt-4.

Model	FewRel				TACRED			
	ACC	**B^3-F_1**	**V-F_1**	**ARI**	**ACC**	**B^3-F_1**	**V-F_1**	**ARI**
Gain prior knowledge from few pre-defined relational instances								
$\text{RoCORE}_{\text{RoBERTa-large}}$	26.85	25.02	54.90	21.79	41.81	39.32	54.76	34.81
$\text{MatchPrompt}_{\text{RoBERTa-large}}$	46.03	46.05	71.18	37.62	52.51	53.02	68.45	45.13
$\text{TABS}_{\text{RoBERTa-large}}$	34.40	31.98	62.81	27.76	43.04	40.45	55.87	35.79
Gain prior knowledge from LLMs								
ChatGPT	13.05	9.94	51.06	7.92	22.81	19.07	33.24	14.43
$\text{ChatGPT}_{\text{+RoBERTa-large}}$	25.27	17.86	39.85	12.25	35.45	28.33	41.49	16.89
$\text{SBERT}_{\text{ChatGPT +RoBERTa-large}}$	33.92	26.52	46.30	17.40	46.31	37.02	54.26	30.61
$\text{CLUSTERLLM}_{\text{ChatGPT +Instructor-XL}}$	47.61	**50.15**	72.73	38.02	54.40	54.89	68.74	47.66
$\text{ORELLM}_{\text{ChatGPT +GPT2-medium}}$	42.06	41.63	68.57	34.11	51.21	51.19	66.52	44.20
$\text{ORELLM}_{\text{ChatGPT +GPT2-large}}$	46.62	43.82	70.83	35.88	54.73	53.21	**69.61**	48.56
$\text{ORELLM}_{\text{ChatGPT +GPT2-XL}}$	**49.53**	47.10	**73.19**	**39.61**	**55.80**	**55.26**	69.07	**49.74**

Table 1: Overall results of the compared models (**K is known**). The top part shows OpenRE methods without LLMs but using few labeled pre-defined relations as prior knowledge (more details are in Appendix A.2). The bottom part shows the unsupervised OpenRE methods that gain prior knowledge from LLMs. For our ORELLM and SBERT, we execute the stop condition with known K. For other baselines, we use the ground truth K as the cluster number.

CLUSTERLLM (Zhang et al., 2023b). For more details, please refer to Appendix A.2.

Metrics and Implementations. We adopt four commonly used metrics to evaluate clustering results: Accuracy (ACC), B^3-F_1 (Bagga and Baldwin, 1998), V-measure F_1 (V-F_1) (Rosenberg and Hirschberg, 2007), and Adjusted Rand Index (ARI) (Hubert and Arabie, 1985). All experiments are conducted on two NVIDIA A800 (2*80GB). More details about metrics and implementations are in Appendix A.3.

3.2 Main Results

The evaluation results are presented in Table 1 and Table 2 for known K and unknown K, respectively. We can make the following observations: **(1) Directly using LLMs for OpenRE does not work well.** For OpenRE, when we directly adopt generative LLMs, such as ChatGPT, the model generates corresponding relational phrases for given sentences. As shown in Table 1, we add a row of results named "ChatGPT", which is calculated by string matching between phrases. As can be observed, these results are notably low. However, simply aligning phrases with the RoBERTa-large encoder significantly improves ACC and ARI by $12.22\%/12.64\%$ and $5.33\%/2.46\%$ on FewRel/TACRED, respectively. This demonstrates that although LLMs can generate various phrases for a relation, they are strongly semantically related. Further suitable processing can help capture and measure these semantic correlations. To achieve this, our ORELLM employs a generative language model to compute similarities for clustering. The results demonstrate that its accuracy outperforms $\text{ChatGPT}_{\text{+RoBERTa-large}}$ by 14.16% to 24.26% across two datasets, regardless of whether K is known or not. **(2) With appropriate strategies, the potential of LLMs on OpenRE can be unleashed.** Moreover, CLUSTERLLM leverages the knowledge in LLMs to annotate unlabeled data, achieving superior clustering results compared to $\text{ChatGPT}_{\text{+RoBERTa-large}}$. Interestingly, compared to ORELLM, although $\text{SBERT}_{\text{ChatGPT +RoBERTa-large}}$ only has the composition of similarity estimation changed, its performance drops significantly. The reason might be that it relies on an additional metric (e.g., cosine in our experiment) to determine similarity after obtaining phrase representations, which causes more errors in getting probability distributions than ORELLM. Also, the training objective of $\text{SBERT}_{\text{ChatGPT +RoBERTa-large}}$ shifts to a classification paradigm (using pseudo labels), which is different from the pre-training objective of language models. This divergence may hinder the effective transfer of knowledge from pre-trained language models to unsupervised clustering. **(3) Our ORELLM achieves competitive results in all performance indicators on both datasets, es-**

Model	FewRel				TACRED			
	ACC	B^3-F_1	V-F_1	ARI	ACC	B^3-F_1	V-F_1	ARI
Gain prior knowledge from few pre-defined relational instances								
$\text{RoCORE}_{\text{RoBERTa-large}}$	22.93	20.86	49.93	17.70	38.12	35.43	46.04	28.61
$\text{MatchPrompt}_{\text{RoBERTa-large}}$	38.54	39.11	68.55	31.69	46.57	45.53	65.70	38.75
$\text{TABS}_{\text{RoBERTa-large}}$	32.75	28.25	58.33	23.95	40.44	36.05	48.39	31.37
Gain prior knowledge from LLMs								
$\text{ChatGPT}_{\text{+RoBERTa-large}}$	23.26	16.43	35.45	9.75	33.19	26.40	40.26	15.60
$\text{SBERT}_{\text{ChatGPT +RoBERTa-large}}$	29.50	23.45	42.70	15.53	42.82	34.65	49.73	26.83
$\text{CLUSTERLLM}_{\text{ChatGPT +Instructor-XL}}$	42.90	**44.27**	70.27	35.37	50.07	46.89	66.38	43.52
$\text{ORELLM}_{\text{ChatGPT +GPT2-medium}}$	40.38	40.11	68.06	33.27	47.35	45.72	65.20	40.10
$\text{ORELLM}_{\text{ChatGPT +GPT2-large}}$	43.44	41.70	68.29	33.64	49.52	47.61	66.20	42.76
$\text{ORELLM}_{\text{ChatGPT +GPT2-XL}}$	**46.03**	43.38	**70.82**	**36.40**	**52.55**	**51.26**	**68.51**	**45.05**

Table 2: Overall results of the compared models (**K is unknown**). The top part of the table shows OpenRE methods without LLMs but using few labeled pre-defined relations as prior knowledge. The bottom part shows the unsupervised OpenRE methods that gain prior knowledge from LLMs (more details are in Appendix A.2). For our ORELLM and SBERT, we execute the stop condition for unknown K. For other baselines, we follow the method in (Zhang et al., 2021a) to estimate K.

pecially when K is unknown. Existing OpenRE methods first obtain representations and then apply clustering algorithms based on these representations. To perform well in clustering, these methods require prior knowledge of the new relation number, i.e., K. However, assuming a known K is unrealistic, as it is unknown and constantly changes in the open world. Our ORELLM leverages a generative language model to measure similarities between instances and is able to automatically identify the number of new relations based on the similarity distribution. When K is unknown, even if we only use the GPT2-medium as a similarity probability estimator, the performance of ACC and ARI exceed SOTA traditional method, i.e., $\text{MatchPrompt}_{\text{RoBERTa-large}}$, by $1.84\%/0.86\%$ and $1.58\%/1.35\%$ on FewRel/TACRED, respectively. Also, the results are competitive compared with CLUSTERLLM, which uses the Instructor with 1.5B parameters for representation learning. These results illustrate the effectiveness and superiority of the proposed ORELLM.

3.3 Analysis

Effectiveness of Rethink. To validate the effectiveness of "Rethink" in the cooperative cycle, we conducted ablation studies by removing it. As illustrated in Table 3, w/o Rethink reduces model performance by 3.26% and 2.20% on average in FewRel and TACRED, respectively. Furthermore, to clearly show the process and impact of Rethink, we provide case studies that compare phrases generated before and after Rethink in Appendix A.5. The results indicate that the LLM-based phrase extractor can produce improved outcomes when implementing Rethink. These results demonstrate that Rethink can first help the phrase extractor in revising and generating more accurate relational phrases, which subsequently enhances clustering performance.

Effectiveness of Fine-tune. From Table 3, we can observe that w/o Fine-tune, model performance on all indicators decreased sharply by $11.50\% \sim 30.13\%$ on two datasets. This demonstrates that fine-tuning plays a crucial role in improving clustering results. It not only groups instances conveying the same relations but also effectively separates instances of different relations, significantly enhancing the model's ability to classify each data point into the correct clusters.

Analysis on the Cycle Round. When K is known, we evaluate clustering results in each cycle round, as shown in Table 4. Compared with the results in initialization, the model performance on the 1^{st} cycle improves significantly. However, the performance gain seems to reach a plateau and even trends downward after a few cycles. For FewRel, our ORELLM is reaching its capacity after the 2^{nd} cycle of learning. For TACRED, as it has a lower number of relation

Dataset	Model	ACC	B^3-F_1	V-F_1	ARI
FewRel	ORELLM	**49.53**	**47.10**	**73.19**	**39.61**
	w/o Rethink	45.09	44.38	70.52	36.39
	Δ	4.44↓	2.72↓	2.67↓	3.22↓
	w/o Fine-tune	31.23	35.60	55.49	16.72
	Δ	18.30↓	11.50↓	17.70↓	19.67↓
TACRED	ORELLM	**55.80**	**55.26**	**69.07**	**49.74**
	w/o Rethink	53.95	52.19	67.83	47.12
	Δ	1.85↓	3.07↓	1.24↓	2.62↓
	w/o Fine-tune	40.75	36.34	53.80	19.61
	Δ	15.05↓	18.92 ↓	15.27 ↓	30.13↓

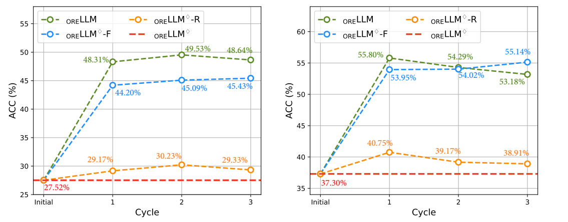
Table 3: Ablation results on FewRel and TACRED. The K is known. The configuration of ORELLM is ChatGPT + GPT2-XL.

Dataset	Cycle	ACC	B^3-F_1	V-F_1	ARI
FewRel	initial	27.52	30.73	52.28	13.30
	1st	$48.31_{+20.79}$	$45.04_{+14.31}$	$71.98_{+19.7}$	$37.62_{+24.32}$
	2nd	$\mathbf{49.53}_{+1.22}$	$\mathbf{47.10}_{+2.06}$	$\mathbf{73.19}_{+1.21}$	$\mathbf{39.61}_{+1.99}$
	3rd	$48.64_{-0.89}$	$45.95_{-1.15}$	$71.07_{-2.12}$	$36.35_{-3.26}$
TACRED	initial	37.30	35.16	51.03	18.78
	1st	$\mathbf{55.80}_{+18.5}$	$\mathbf{55.26}_{+20.10}$	$\mathbf{69.07}_{+18.04}$	$\mathbf{49.74}_{+30.96}$
	2nd	$54.29_{-1.51}$	$53.74_{-1.52}$	$68.18_{-0.89}$	$48.52_{-1.22}$
	3rd	$53.18_{-1.11}$	$52.58_{-1.16}$	$67.82_{-0.36}$	$48.29_{-0.23}$

Table 4: Clustering results of $\text{ORELLM}_{\text{ChatGPT+ GPT2-XL}}$ in each cycle round. The K is known. The subscripted numbers indicate the difference between the current cycle and its last one.

types than FewRel, the ORELLM achieves a good performance just after the 1st cycle. To explore reasons for this phenomenon, we represent the model with neither Rethink nor Fine-tune in initial cycle as ORELLM$^\Diamond$, and models with only Rethink or Fine-tune as ORELLM$^\Diamond$-R and ORELLM$^\Diamond$-F, respectively. We compare the performance among ORELLM, ORELLM$^\Diamond$, ORELLM$^\Diamond$-R and ORELLM$^\Diamond$-F in each cycle round and the results are illustrated in Figure 3.

Observing the trend from the initial state to the 3rd cycle, we note that ORELLM, ORELLM$^\Diamond$-F, and ORELLM$^\Diamond$-R exhibit the most substantial performance gains in the 1st cycle. In subsequent cycles, the improvements tend to stabilize, indicating that one cycle round is sufficient for the models to achieve better performance. Specifically, the performance of ORELLM$^\Diamond$-R decreases after the 1st cycle, which suggests that Rethink just one time is suitable for ORELLM. This is likely because Rethink many times leads ChatGPT to generate rare and unconventional phrases as it tries to avoid repetition of all previously generated phrases for an instance. These unconventional phrases may introduce noise, resulting in a downward trend in the performance of ORELLM.

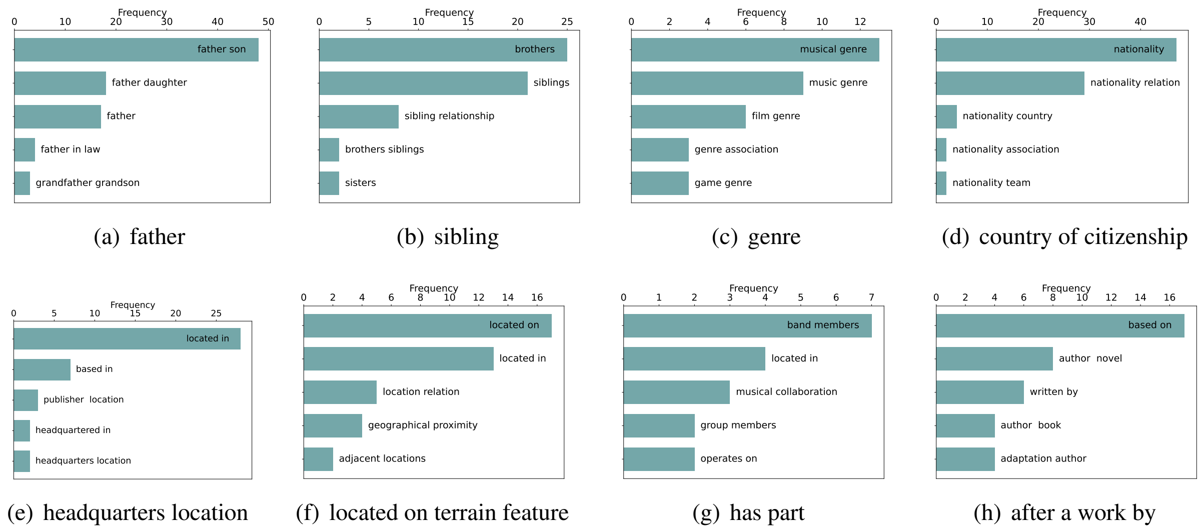


(a) FewRel (b) TACRED

Figure 3: Clustering accuracy (ACC) of different variant models in different cycles. (i) ORELLM: original model; (ii) ORELLM$^\Diamond$-R: the model only with Rethink; (iii) ORELLM$^\Diamond$-F: the model only with Fine-tune; (iiii) ORELLM$^\Diamond$: the model with neither Rethink nor Fine-tune; The configuration of every model is ChatGPT + GPT2-XL.

3.4 Which relation can be well recognized by generative LLMs?

We use the new relations in FewRel to explore which relations can be well recognized by LLMs. We first assess the accuracy of each relation and select four relations with higher accuracy (e.g., *father*, *sibling*, *genre*, and *country of citizenship*) and four with lower (e.g., *headquarters location*, *located on terrain feature*, *has part*, and *after a work by*) for further analysis. For each relation, we count the top 5 most frequent phrases generated from 70 instances. As illustrated in Figure 4, the observations are as follows: **(1) The rich background knowledge of LLMs makes them adept at identifying common relations.** For instance, they can accurately produce relational phrases such as “father-son, father” and “sibling relationships, brothers” for family relations *father* and *sibling*, respectively. Moreover, this knowledge also allows them to identify implicit relations. In relation *genre*, for a given instance “Trailer Bride was a Chapel Hill, North Carolina-based alternative country rock band signed to Bloodshot Records” with “Trailer Bride” and “alternative country rock” as the head and tail entities, LLMs are capable of generating the relational phrase “genre affiliation”, even if the term “genre” is not explicitly mentioned in this sentence. **(2) It is difficult for LLMs to distinguish the granularity of relations.** *Headquarters location* and *located on terrain feature* are different relations in FewRel, yet both are related to locations. Without specific prompts that focus on their differences, LLMs cannot distinguish granularities between these relations and tend to produce generic phrases like “located in, located on” for the vast



(a) father (b) sibling (c) genre (d) country of citizenship

(e) headquarters location (f) located on terrain feature (g) has part (h) after a work by

Figure 4: The top 5 most frequent phrases in different categories. Subgraphs (a)-(d) and (e)-(h) represent relations with higher and lower accuracy, respectively. The description of these relations are in Appendix A.6.

majority of instances (as shown in Figure 4 (e) and (f)). Similarly, for coarser-grained relations like *has part*, LLMs often generate fine-grained phrases such as “group member, season of, band member” according to the context of sentences.

4 Related Works

Open Relation Extraction. Methods for Open relation extraction (OpenRE) can be categorized as: tagging-based (Banko et al., 2007; Yates et al., 2007) and clustering-based (Hu et al., 2020; Simon et al., 2019; ElSahar et al., 2017). Tagging-based methods directly extract relation-related phrases from sentences based on syntactic analysis, but often produce redundant or ambiguous results (Fader et al., 2011). Therefore, clustering-based methods have drawn more attention. Liu et al. (2021) revisits OpenRE from a causal view and Zhang et al. (2021b) incorporates hierarchical information into this task. Additionally, Zhao et al. (2021) learns a relational-oriented clustering model and Wang et al. (2022) proposes a prompt-based framework to identify open relations. Li et al. (2022) introduces the idea of introduce the idea of type abstraction and learns novel relations from two views. Zhao et al. (2023) integrates the active learning with clustering, which improves the clustering performance while discovering as many relations as possible. In this work, we propose a novel method for OpenRE based on large language models.

Relation Extraction with Large Language Models. Recently, using large language models (LLMs) for relation extraction (RE) has received much attention (Wadhwa et al., 2023; Wan et al., 2023; Ding et al., 2024). Wan et al. (2023) enriches the reasoning evidence of each demonstration for RE. Zhang et al. (2023a) unlocks the power of LLMs as zero-shot relation extractors by reformulating RE into multiple-choice question answering. Li et al. (2023) introduces a novel prompting method to fully explore the power of LLMs, thereby improving their performance on zero-shot RE. Most of these works are focused on zero-shot or few-shot RE, and OpenRE still needs to be explored. We propose a new idea for OpenRE in the era of LLMs, that is, first extract relational phrases and then directly exploit the knowledge in LLMs to assess the semantic similarity between phrases, without relying on any additional metrics.

5 Conclusion

In this work, we propose a new idea for OpenRE in the era of LLMs: extracting relational phrases and directly exploiting the knowledge in LLMs to assess the semantic similarity between phrases without relying on any additional metrics. Based on this idea, we develop a framework named ORELLM, which consists of two LLMs. One serves as a phrase extractor, while the other operates as a similarity probability estimator. These two LLMs work together through a cooperative cycle for clustering. Experimental results show that ORELLM outperforms current baselines by $1.4\% \sim 3.13\%$ in terms of clustering accuracy.

Limitations

We consider that our current method has the following two limitations. (1) LLMs have hallucinations and may produce incorrect phrases. Although we perform "Rethink" to reduce this problem, it still exists. Therefore, we need to design a more efficient strategy to allow the model to generate more effective relational phrases. (2) Since the probability of generation is calculated by multiplying the likelihood of each token in the sentence conditioned on all its previous tokens, the result will be smaller as the phrase becomes longer (even in log domain). This makes it difficult to distinguish between phrases if all calculated similarities are small. In our work, we limit the length of generated relational phrases to no more than five words. However, five words may not accurately describe some relations, especially when distinguishing fine-grained ones, e.g., *located in the administrative territorial entity* and *located on terrain feature*.

Acknowledgments

We would like to thank all anonymous reviewers for their constructive comments. This work was supported by National Key Research and Development Program of China (2022YFC3303600), National Natural Science Foundation of China (62137002, 62293553, 62176207 and 62192781), "LENOVO-XJTU" Intelligent Industry Joint Laboratory Project, Natural Science Basic Research Program of Shaanxi (2023-JC-YB-593), the Youth Innovation Team of Shaanxi Universities, XJTU Teaching Reform Research Project "Acquisition Learning Based on Knowledge Forest", and the National Research Foundation (NRF) Singapore and DSO National Laboratories Singapore under the AI Singapore Program (No. AISG2-RP-2020-016).

References

David Arthur and Sergei Vassilvitskii. 2007. k-means++: the advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2007, New Orleans, Louisiana, USA, January 7-9, 2007*, pages 1027–1035.

Amit Bagga and Breck Baldwin. 1998. Algorithms for scoring coreference chains. In *The first international conference on language resources and evaluation workshop on linguistics coreference*, pages 563–566.

Michele Banko, Michael J. Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007*, pages 2670–2676.

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*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Zepeng Ding, Wenhao Huang, Jiaqing Liang, Yanghua Xiao, and Deqing Yang. 2024. Improving recall of large language models: A model collaboration approach for relational triple extraction. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, LREC/COLING 2024, 20-25 May, 2024, Torino, Italy*, pages 8890–8901.

Hady ElSahar, Elena Demidova, Simon Gottschalk, Christophe Gravier, and Frédérique Laforest. 2017. Unsupervised open relation extraction. In *The Semantic Web: ESWC 2017 Satellite Events - ESWC 2017 Satellite Events, Portorož, Slovenia, May 28 - June 1, 2017, Revised Selected Papers*, volume 10577 of *Lecture Notes in Computer Science*, pages 12–16.

Oren Etzioni, Michele Banko, Stephen Soderland, and Daniel S. Weld. 2008. Open information extraction from the web. *Commun. ACM*, 51(12):68–74.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1535–1545.

Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. 2019. Representation degeneration problem in training natural language generation models. In

7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018. Fewrel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 4803–4809.

Xuming Hu, Lijie Wen, Yusong Xu, Chenwei Zhang, and Philip S. Yu. 2020. Selfore: Self-supervised relational feature learning for open relation extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 3673–3682.

Lawrence Hubert and Phipps Arabie. 1985. Comparing partitions. *Journal of classification*, 2(1):193–218.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the sentence embeddings from pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 9119–9130.

Guozheng Li, Peng Wang, and Wenjun Ke. 2023. Revisiting large language models as zero-shot relation extractors. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 6877–6892.

Sha Li, Heng Ji, and Jiawei Han. 2022. Open relation and event type discovery with type abstraction. 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 6864–6877.

Fangchao Liu, Hongyu Lin, Xianpei Han, Boxi Cao, and Le Sun. 2022. Pre-training to match for unified low-shot 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 5785–5795.

Fangchao Liu, Lingyong Yan, Hongyu Lin, Xianpei Han, and Le Sun. 2021. Element intervention for open relation extraction. 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 4683–4693.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

Leland McInnes, John Healy, and Steve Astels. 2017. hdbscan: Hierarchical density based clustering. *J. Open Source Softw.*, 2(11):205.

G Ayorkor Mills-Tettey, Anthony Stentz, and M Bernardine Dias. 2007. The dynamic hungarian algorithm for the assignment problem with changing costs. *Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-07-27*.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. 2020. Pre-trained models for natural language processing: A survey. *CoRR*, abs/2003.08271.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Emily Reif, Ann Yuan, Martin Wattenberg, Fernanda B. Viégas, Andy Coenen, Adam Pearce, and Been Kim. 2019. Visualizing and measuring the geometry of BERT. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8592–8600.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 3980–3990.

Andrew Rosenberg and Julia Hirschberg. 2007. V-measure: A conditional entropy-based external cluster evaluation measure. In *EMNLP-CoNLL 2007, Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 410–420.

Étienne Simon, Vincent Guigue, and Benjamin Piwowarski. 2019. Unsupervised information extraction: Regularizing discriminative approaches with relation distribution losses. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 1378–1387.

Xiaoshuai Song, Keqing He, Pei Wang, Guanting Dong, Yutao Mou, Jingang Wang, Yunsen Xian, Xunliang Cai, and Weiran Xu. 2023. Large language models meet open-world intent discovery and recognition: An evaluation of chatgpt. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 10291–10304.

Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. 2023. One embedder, any task: Instruction-finetuned text embeddings. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 1102–1121.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971.

Thy Thy Tran, Phong Le, and Sophia Ananiadou. 2020. Revisiting unsupervised relation extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7498–7505.

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.

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.

Chengyi Wang, Yu Wu, Shujie Liu, Zhenglu Yang, and Ming Zhou. 2020. Bridging the gap between pre-training and fine-tuning for end-to-end speech translation. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 9161–9168.

Jiaxin Wang, Lingling Zhang, Jun Liu, Xi Liang, Yujie Zhong, and Yaqiang Wu. 2022. Matchprompt: Prompt-based open relation extraction with semantic consistency guided clustering. 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 7875–7888.

Benfeng Xu, Quan Wang, Yajuan Lyu, Dai Dai, Yongdong Zhang, and Zhendong Mao. 2023. S2ynre: Two-stage self-training with synthetic data for low-resource relation extraction. 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 8186–8207.

Zhaohui Yan, Songlin Yang, Wei Liu, and Kewei Tu. 2023. Joint entity and relation extraction with span pruning and hypergraph neural networks. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 7512–7526.

Limin Yao, Aria Haghighi, Sebastian Riedel, and Andrew McCallum. 2011. Structured relation discovery using generative models. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1456–1466.

Alexander Yates, Michele Banko, Matthew Broadhead, Michael J. Cafarella, Oren Etzioni, and Stephen Soderland. 2007. Textrunner: Open information extraction on the web. In *Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, April 22-27, 2007, Rochester, New York, USA*, pages 25–26.

Hanlei Zhang, Hua Xu, Ting-En Lin, and Rui Lyu. 2021a. Discovering new intents with deep aligned clustering. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 14365–14373.

Kai Zhang, Bernal Jimenez Gutierrez, and Yu Su. 2023a. Aligning instruction tasks unlocks large language models as zero-shot relation extractors. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 794–812.

Kai Zhang, Yuan Yao, Ruobing Xie, Xu Han, Zhiyuan Liu, Fen Lin, Leyu Lin, and Maosong Sun. 2021b. Open hierarchical relation extraction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 5682–5693.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 35–45.

Yuwei Zhang, Zihan Wang, and Jingbo Shang. 2023b. Clusterllm: Large language models as a guide for text clustering. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 13903–13920.

Jun Zhao, Tao Gui, Qi Zhang, and Yaqian Zhou. 2021. A relation-oriented clustering method for open relation extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 9707–9718.

Jun Zhao, Yongxin Zhang, Qi Zhang, Tao Gui, Zhongyu Wei, Minlong Peng, and Mingming Sun. 2023. Actively supervised clustering for open relation extraction. 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 4985–4997.

A Appendix

A.1 Models

We adopt ChatGPT [4](`gpt-3.5-turbo`) or GPT-4 [5](`gpt-4`) as our phrase extractor. Considering the balance between cost and performance, we first use ChatGPT to extract relational phrases for all input instances. Then, we employ GPT-4 to apply the "Rethink" on ambiguous data. As for probability estimator, we utilize a publicly available GPT2-XL model (Radford et al., 2019) with 1.5B parameters. To better evaluate the efficiency and flexibility of our framework, two smaller generative models: GPT2-medium and GPT2-large, each with 355M and 774M parameters respectively, are also adopted as probability estimators.

A.2 Baselines

To demonstrate the effectiveness of our method, we compare it with the following six OpenRE baselines, which are categorized into two groups: (1) Three traditional methods with pre-defined labeled instances for pre-training: **RoCORE** (Zhao et al., 2021), **MatchPrompt** (Wang et al., 2022), and **TABS** (Li et al., 2022). These methods leverage labeled pre-defined relational instances as prior knowledge to help discover new relations and achieve state-of-the-art results. We construct the labeled data for these methods by randomly selecting 16 instances of each type from the remaining 16 and 14 relations in FewRel and TACRED, respectively. (2) Three methods using LLMs: **ChatGPT** $_{\textbf{+RoBERTa-large}}$, we adopt RoBERTa-large (355M) to learn the embedding of relational phrases generated by ChatGPT and perform the clustering algorithm K-Means (Arthur and Vassilvitskii, 2007); **SBERT** $_{\textbf{ChatGPT +RoBERTa-large}}$, we use SentenceBERT (Reimers and Gurevych, 2019), which estimate pairwise similarity between phrases by siamese BERT, and keep the remaining model structure consistent with our oreLLM; **ClusterLLM** (Zhang et al., 2023b), which leverages ChatGPT or GPT-4 to annotate data and guide a pre-trained embedder named Instructor (Su et al., 2023) for text clustering. We change the embedder to Instructor-XL (1.5B) for comparison.

A.3 Metrics and Implementations

We adopt four commonly used metrics to evaluate clustering results: Accuracy (ACC), B^3-F_1 (Bagga and Baldwin, 1998), V-measure F_1 (V-F_1) (Rosenberg and Hirschberg, 2007), and Adjusted Rand Index (ARI) (Hubert and Arabie, 1985). Specifically, ACC is calculated using the Hungarian algorithm (Mills-Tettey et al., 2007) to evaluate the best mapping between clustering assignments and true labels; B^3-F_1 is the harmonic mean of precision and recall for placing each instance in its cluster; Similarly, V-F_1 represents the harmonic mean of homogeneity and completeness for clusters; ARI reflects the degree of agreement between clusters and true distributions.The probability estimator fine-tunes with a learning rate of $3e-5$, $2e-5$, and $2e-5$ for GPT2-medium, GPT2-large, and GPT-XL by AdamW (Loshchilov and Hutter, 2019), respectively. P_ϵ and P_r are given by the value of elements ranked top 40% and 20% in P from large to samll, respectively. Threshold N_ε is 20 for both datasets. The γ are 0.7 and 0.8 for FewRel and TACRED, respectively. We set the temperatures as 0.5 and 0.8 for `gpt-3.5-turbo` and `gpt-4`, respectively. The generated lengths for $\{p_1, p_3\}$ and $\{p_2, p_4\}$ are 256 and 16, respectively.

A.4 Prompts for LLMs

We list all templates used in our work including $\{p_1, p_2\}$ to extract relational phrases and $\{p_3, p_4\}$ to rethink as follows:

p_1: `In sentence {`s_i`}, the head entity is {`h_i`} and the tail entity is {`t_i`}. please first analyze the type of head entity and the tail entity, then describe the relation between them. [Demonstration]`

p_2: `Based on the following analysis, summarize the relation for the head entity and tail entities within five words. To make the answer more specific, you may add entity type into the answer. [Demonstration], {`s_i, h_i, t_i`}, Analysis:{response from `p_1`}.Answer:`

p_3: `Given a sentence {`s_i`}, the corresponding head entity is {`h_i`} and the tail entity is {`t_i`}. You have previously summarized the relation between them as {`w_j`}. However, these phrases might not accurately describe the relation between {`h_i`} and {`t_i`}. Please re-analyze this relation.`

p_4: `Based on the analysis {response from `p_3`}, please rethink and provide a revised phrase to describe the relation between head entity {`h_i`} and tail entity {`t_i`} in`

[4]https://chat.openai.com/chat.

[5]https://chatgpt.com/?oai-dm=1&model=gpt-4.

`sentence {`s_i`}. No more than five words.`

As shown in Table 5, we give an example of using p_1 and p_2 to generate relational phrases.

A.5 Case Studies on Rethink

To clearly show the process and impact of Rethink, we provide case studies that compares phrases generated before and after Rethink. The results are illustrated in Table 6 and Table 7. These results indicate that the LLM-based phrase extractor can produce improved outcomes when doing Rethink.

A.6 Description of Relations

We list the description of relations that appear in Section 3.4 as follows[6] [7]:

(a) *father*: male parent of the subject.

(b) *sibling*: the subject has the object as their sibling (brother, sister, etc.).

(c) *genre*: creative work's genre or an artist's field of work.

(d) *country of citizenship*: the object is a country that recognizes the subject as its citizen.

(e) *headquarters location*: specific location where an organization's headquarters is or has been situated.

(f) *located on terrain feature*: located on the specified landform. Should not be used when the value is only political or administrative (provinces, states, countries, etc.).

(g) *has part*: part of this subject.

(h) *after a work by*: artist whose work strongly inspired or was copied in this item.

[6]The description of other relations for FewRel are in https://github.com/thunlp/FewRel/fewrel.pdf

[7]The description of relations in TACRED are in https://nlp.stanford.edu/projects/tacred

Prompt p_1	In sentence {The Breaks is an American 1999 comedy film written by and starring Mitch Mullany.}, the head entity is {Mitch Mullany} and the tail entity is {American}. please first analyze the type of head entity and the tail entity, then describe the relation between them. `[For example, The occupancy of the show surpassed the Hindi films Baar Baar Dekho and Freaky Ali., {h}: Baar Baar Dekho, {t}: Hindi. {Analysis}: Baar Baar Dekho is mentioned as a film. The Hindi here is referring to the language. The relation between them means the language in which the film is made.]`
Response	Mitch Mullany is a notable figure within the context of the film. Therefore, the relation between “Mitch Mullany” and “American” indicates his involvement in the American comedy film, possibly as a key figure in its production or performance.
Prompt p_2	Based on the following analysis, summarize the relation for the head entity and tail entities within five words. To make the answer more specific, you may add entity type into the answer. `[For example, The occupancy of the show surpassed the Hindi films Baar Baar Dekho and Freaky Ali., {h}: Baar Baar Dekho, {t}: Hindi. {Analysis}: Baar Baar Dekho is mentioned as a film. The Hindi here is referring to the language. The relation between them means the language in which the film is made. Answer: language of film.]` {The Breaks is an American 1999 comedy film written by and starring Mitch Mullany.}, {h}: {Mitch Mullany}, {t}:{American}, Analysis:{Mitch Mullany is...}. Answer:
Response	writer of film

Table 5: An example of generating relational phrases.

Ground Truth Relation: *Country of citizenship* Example #1	
Prompt p_3	Given a sentence {Heinz, legal name Mathias Halfpape (born in Hamburg) is a German entertainer.}, the corresponding head entity is {Mathias Halfpape} and the tail entity is {German}. You have previously summarized the relation between them as {**legal name of person**}. However, these phrases might not accurately describe the relation between {Mathias Halfpape} and {German}. Please re-analyze this relation.
Response	The "Mathias Halfpape" as a legal name but also adds that he is a "German entertainer". The term "German" here relates to nationality or citizenship rather than being part of a legal name. Therefore, a more appropriate relation between "Mathias Halfpape" and "German" in this context would be "nationality" or "citizenship", not "legal name".
Prompt p_4	Based on the analysis{The "Mathias Halfpape"...}, please rethink and provide a revised phrase to describe the relation between head entity {Mathias Halfpape} and tail entity {German} in sentence {Heinz, legal name Mathias Halfpape (born in Hamburg) is a German entertainer.} No more than five words.
Response	**nationality of person**
Ground Truth Relation: *Country of citizenship* Example #2	
Prompt p_3	Given a sentence {The Breaks is an American 1999 comedy film written by and starring Mitch Mullany.}, the corresponding head entity is {Mitch Mullany} and the tail entity is {American}. You have previously summarized the relation between them as {**writer of film**}. However, these phrases might not accurately describe the relation between {Mitch Mullany} and {American}. Please re-analyze this relation.
Response	The term "American" in this sentence refers to the nationality or origin of the film, not a person or object Mitch Mullany wrote. As"Mitch Mullany" is a person name, the relation between "Mitch Mullany" and "American" should be analyzed in terms of nationality association.
Prompt p_4	Based on the analysis{The term"American"...}, please rethink and provide a revised phrase to describe the relation between head entity {Mitch Mullany} and tail entity {American} in sentence {The Breaks is an American 1999 comedy film written by and starring Mitch Mullany.} No more than five words.
Response	**nationality of Mitch Mullany**

Table 6: Case study of Rethink on the ground truth relation *Country of citizenship* in FewRel.

Ground Truth Relation: *org: website*	Example #1
Prompt p_3	Given a sentence {Event Sponsors: West Marine <http://docs.google.com/www.westmarine.com> }, the corresponding head entity is {West Marine} and the tail entity is {<http://docs.google.com/www.westmarine.com>}. You have previously summarized the relation between them as {**sponsor of** }. However, these phrases might not accurately describe the relation between {West Marine} and {<http://docs.google.com/www.westmarine.com>}. Please re-analyze this relation.
Response	If the URL is intended to point to a webpage hosting details about West Marine's sponsorship activities or an event they sponsor, then West Marine is not sponsoring the webpage itself but is the subject or featured sponsor on that page. Therefore, the link points to a webpage about West Marine's sponsorship rather than indicating that West Marine sponsors the webpage itself.
Prompt p_4	Based on the analysis{If the URL...}, please rethink and provide a revised phrase to describe the relation between head entity {West Marine} and tail entity {<http://docs.google.com/www.westmarine.com>} in sentence {Event Sponsors: West Marine <http://docs.google.com/www.westmarine.com>.} No more than five words.
Response	**website of sponsor**

Ground Truth Relation: *org: website*	Example #2
Prompt p_3	Given a sentence {Guide to ABC Radio Australia (RA) ABC RA = >http://www.abc.net.au/ra/guide/. }, the corresponding head entity is {Guide to ABC Radio Australia} and the tail entity is { http://www.abc.net.au/ra/guide/}. You have previously summarized the relation between them as {**CEO of** }. However, these phrases might not accurately describe the relation between {Guide to ABC Radio Australia} and {http://www.abc.net.au/ra/guide/}. Please re-analyze this relation.
Response	The "Guide to ABC Radio Australia" indicates a document, webpage, or resource providing information or directions regarding ABC Radio Australia.The URL "http://www.abc.net.au/ra/guide/" likely represents the actual location where such a guide can be accessed or viewed online.
Prompt p_4	Based on the analysis{The "Guide to ABC Radio Australia" indicates...}, please rethink and provide a revised phrase to describe the relation between head entity {Guide to ABC Radio Australia} and tail entity {http://www.abc.net.au/ra/guide/} in sentence {Guide to ABC Radio Australia (RA) ABC RA = >http://www.abc.net.au/ra/guide/.} No more than five words.
Response	**website link**

Table 7: Case study of Rethink on the ground truth relation *org: website* in TACRED.