LLM4RE: A Data-centric Feasibility Study for Relation Extraction

Anushka Swarup[1], Tianyu Pan[1], Ronald Wilson[1], Avanti Bhandarkar[1], Damon L. Woodard[1]

[1]Florida Institute for National Security (FINS), University of Florida, Gainesville FL 32611, USA.

Correspondence: aswarup@ufl.edu

Abstract

Relation Extraction (RE) is a multi-task process that is a crucial part of all information extraction pipelines. With the introduction of the generative language models, Large Language Models (LLMs) have showcased significant performance boosts for complex natural language processing and understanding tasks. Recent research in RE has also started incorporating these advanced machines in their pipelines. However, the full extent of the LLM's potential for extracting relations remains unknown. Consequently, this study aims to conduct the first feasibility analysis to explore the viability of LLMs for RE by investigating their robustness to various complex RE scenarios stemming from data-specific characteristics. By conducting an exhaustive analysis of five state-of-the-art LLMs backed by more than 2100 experiments, this study posits that LLMs are not robust enough to tackle complex data characteristics for RE, and additional research efforts focusing on investigating their behaviors at extracting relationships are needed. The source code for the evaluation pipeline can be found on here[1].

1 Introduction

Relation Extraction (RE) in Natural Language Processing (NLP) deals with extracting relationships between target nouns or entities from textual data. It is a critical step in Information Extraction (IE) due to its wide-scale applicability for downstream applications such as Knowledge Base (KB) creation and Question Answering (QA).

With the introduction of generative large language models (LLMs) such as GPT-3 (Brown, 2020) and Llama (Touvron et al., 2023), research in RE has also shifted to incorporating such advanced technologies to extract semantic relationships. LLMs are pre-trained on massive amounts of data and exhibit enhanced language understanding capabilities. This prowess has made them a natural choice for complex IE tasks. Although the existing research community has readily adapted to using LLMs for RE (Wan et al., 2023; Xu et al., 2023), their capabilities at extracting relations in the presence of complex data attributes have yet to be thoroughly investigated (Li et al., 2023; Ma et al., 2023).

Issues such as fine-grained and similar relation types, multiple relations and overlapping entities, and scarcity of annotated data have long challenged traditional relation extractors (Aydar et al., 2020; Swarup et al., 2024). These issues arise from the complex nature of data and are abundant in fields such as business, finance, and medicine, where numerous entities interact to form an ecosystem. For example, the sharing of entities between multiple relations has been a challenging use case for most traditional relation extractors as it leads to ambiguous use cases. It is imperative to build relation extractors that are robust to such complex characteristics to apply them to real-world use cases. LLMs, due to the vast knowledge contained in them and the ability for common-sense reasoning, can present a viable solution to efficiently tackle these problems. Thus, this study aims to answer whether LLMs are robust at extracting relationships in the presence of such complex data characteristics. The contributions of this study can be summarised as follows:

- First feasibility study for RE that aims to analyze state-of-the-art (SOTA) LLM families such as GPT, Llama, Mistral, OpenChat, and Gemma based on their capabilities of extracting relationships from complex data.

- Highlights the challenges faced by the current models and enlists insights for future research backed by more than 2100 experiments.

[1] https://aaig.ece.ufl.edu/projects/relation-extraction

Proceedings of the 31st International Conference on Computational Linguistics, pages 6670–6691
January 19–24, 2025. ©2025 Association for Computational Linguistics

2 Related Work

A brief overview of the research endeavors associated with RE, from traditional RE algorithms to generative approaches, has been presented below. The discussion not only details the flow of research in this field but also highlights the various challenges faced by the relation extractors.

2.1 Traditional Approaches for RE

A plethora of research exists in RE that heavily employs neural networks. Early approaches focused on relation classification (RC) and used CNNs and LSTMs (Lee et al., 2019; Zhou et al., 2016) to extract contextual representations of the input that could be used as feature representations for a classifier. With the introduction of the transformer architecture, research shifted solely to finetuning pre-trained language models (PLMs) for RC (Wu and He, 2019; Zhou and Chen, 2022). Additionally, the importance of entities in the extraction process was emphasized, and many approaches aimed to incorporate entity-specific knowledge into the classification process (Yamada et al., 2020; Zhang et al., 2019, 2017a). Although the PLMs provided significant performance gains for RC, it was found that such algorithms were not robust to complex scenarios such as long-tail distribution of data, presence of ambiguous context and relations, and overlapping entities (Swarup et al., 2024; Han et al., 2020).

Subsequently, the focus of the community shifted to the domain of joint entity and relation extraction (JRE), which dealt with the extraction of entities and relations in the form of triplets in a single pipeline. The combined extraction helped seamlessly share important entity-specific information from the entity to the relation extraction phases (Li et al., 2021; Sui et al., 2020; Tang et al., 2022). This paradigm was especially valuable for tackling the problem of overlapping entities (Zhao et al., 2021) as it could extract multiple triplets for the same text sample. Although JRE algorithms were a natural solution for many of the problems faced by their RC counterparts, they were found to be extremely brittle to varying data characteristics. Additionally, the algorithms were prone to high false negative values due to the discrepancies in the semantic structure of ground truth and predicted labels (Wang et al., 2020).

2.2 Generative Approaches for RE

With the introduction of GPT-2 (Radford et al., 2019), the concept of prompt-based RE algorithms became popular. Initial algorithms focused on finding optimal prompts to be used as input to a PLM (Han et al., 2022; Chen et al., 2022). Subsequently, with the introduction of chat-based models, the domain of RE has moved to the use of generative models for classification and extraction. LLMs are trained on vast amounts of data and can make efficient decisions by looking at a few data points during inference. Thus, the potential of LLMs as few-shot or low-resource extractors has been widely studied. However, the consensus remains uncertain. Some studies suggest that LLMs efficiently extract relationships under few-shot settings (Wei et al., 2023). In contrast, others present a contrary belief and support that the LLMs should be used to aid traditional extractors (Ma et al., 2023). Finally, another major issue with using LLMs for RE, similar to traditional JRE, is the difficulty in accurate evaluations. It has been highlighted that the open-ended nature of generative models results in predictions that, although semantically accurate, might differ from the ground truth labels. This issue leads to high false negatives and has become a major concern as without proper evaluation techniques, it is difficult to analyze the capabilities of using LLMs for RE (Wadhwa et al., 2023). To alleviate this issue, the GenRES (Jiang et al., 2024) evaluation benchmark was introduced recently, which aims to qualitatively analyze LLM-based relation predictions.

3 Methodology

The research in RE has been segregated into two paradigms: Relation Classification (RC) and Joint Relation Extraction (JRE). RC can be defined as extracting a relationship r between two entities $e1$ and $e2$ such that $r \in R$ where R is a set of predefined relation types. On the other hand, JRE can be described as extracting entity relation triplets $(entity1, relation, entity2)$ from text when the entity information is unknown before extraction. This study takes a two-pronged approach and explores both paradigms to achieve a holistic view of the problem.

Although LLMs have been shown to provide significant performance gains, especially in low-resource settings, it has been observed that the LLM's performance is not stable and is usually

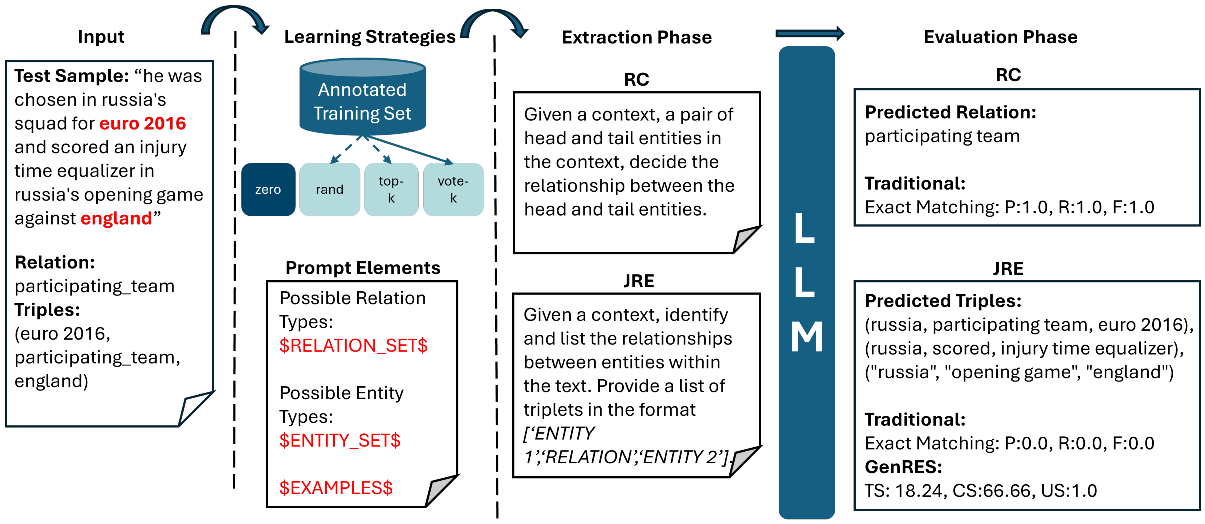

Input
Learning Strategies
Extraction Phase
Evaluation Phase
Test Sample: "he was chosen in russia's squad for euro 2016 and scored an injury time equalizer in russia's opening game against england"
Relation:
participating_team
Triples:
(euro 2016, participating_team, england)
Annotated Training Set
zero
rand
top-k
vote-k
Prompt Elements
Possible Relation Types:
$RELATION_SET$
Possible Entity Types:
$ENTITY_SET$
$EXAMPLES$
RC
Given a context, a pair of head and tail entities in the context, decide the relationship between the head and tail entities.
JRE
Given a context, identify and list the relationships between entities within the text. Provide a list of triplets in the format ['ENTITY 1','RELATION','ENTITY 2']
LLM
RC
Predicted Relation:
participating team
Traditional:
Exact Matching: P:1.0, R:1.0, F:1.0
JRE
Predicted Triples:
(russia, participating team, euro 2016),
(russia, scored, injury time equalizer),
("russia", "opening game", "england")
Traditional:
Exact Matching: P:0.0, R:0.0, F:0.0
GenRES:
TS: 18.24, CS:66.66, US:1.0

Figure 1: Evaluation pipeline for LLM-based RC and JRE inference.

highly susceptible to inference conditions. Some such techniques include the choice of learning strategy (zero-shot vs few-shot) and prompt construction. Thus, to fully investigate LLMs, this study examines their performance under various inference conditions. The overall evaluation pipeline can be divided into three key components: 1) learning strategy, 2) prompt engineering, and 3) evaluation. Figure 1 depicts the overall pipeline, and more details of each strategy are discussed below.

3.1 Learning Strategy

The first phase of the evaluation pipeline deals with the strategy used to help the LLM learn the target task. Although LLMs can perform tasks in low-resource settings such as zero-shot, it has been observed that the LLM's performance increases many-fold when few training samples are provided during inference (Wei et al., 2022). To achieve this, in-context learning (ICL) has emerged as one of the most popular strategies where an LLM is given knowledge of the task through selective training samples and labels that act as demonstrations (Dong et al., 2022). This study employs three popular ICL strategies to retrieve demonstrations from training data along with zero-shot inference. The details are as follows:

- Zero-shot Inference (**zero**): Only the test data is provided as input via a prompt without incorporating any training samples as demonstrations.
- Random retriever (**random**) - randomly selects K training samples to be used as demonstrations for the LLM.
- KNN-based retriever (**topK**) - extracts K most semantically similar samples to the test samples. The semantic similarity is calculated using BERT-based sentence embeddings, and nearest neighbors are found using a KNN (Wan et al., 2023; Liu et al., 2022).
- Diversity-based retriever (**voteK**) - a two-stage process that uses graph-based techniques to extract N selective samples from the training set. This smaller pool is then used to select K samples using a prompt-retrieval strategy optimized using the context length of the prompt. The *fast-votek* implementation of the work was used for this study (Su et al., 2022).

3.2 Prompt-Engineering

Next, prompts were carefully curated with the RC and JRE tasks in mind. Two key variables when creating prompts for RE can be the incorporation of relation label verbalizations and entity-type information in the prompts. Thus, based on the past literature (Jiang et al., 2024; Wadhwa et al., 2023), the following prompting strategies were used for LLM inference:

- No additional information (**open**): used to analyze the performance of the LLM in an open-ended setting where no prior information about the target relation labels and entity types has been provided. Used for both RC and JRE.

- Only relation information (**rel++**): used to analyze RE capabilities of the LLMs when the target relation set is provided in the prompt using label verbalizations. Used only for RC.
- Only entity type information (**ent++**): used to study the influence of prior knowledge of entity type information on the RE process. Only used for RC where the entity type information was incorporated within each context sample.
- Both entity and relation type information (**entrel++**): a combination of rel++ and ent++ prompts used to analyze the performance of the LLMs when maximum auxiliary information is provided. Used for both RC and JRE.

Prompt templates for all the above-mentioned categories can be found in the Appendix A.7.

3.3 Evaluation Strategies

The final step in the pipeline is the evaluation phase. As discussed above, this study aims to investigate LLMs and their performance at extracting relationships when complex data characteristics are present. Thus, the following data attributes were investigated as part of this study:

- *Fine-grained Datasets* deal with the complexity that arises when the label space contained in a dataset is large. Larger label spaces can lead to multiple relations having similar meanings, which might confuse the models and cause them to make incorrect predictions.
- *Multiple relations & Overlapping entities* deal with the presence of multiple relation types in a text sample. This scenario often occurs in conjunction with the overlap of a single or pair of entities with the relations.
- *Low resource scenarios* deal with the low availability of annotated data that can help the model learn the distribution of different relation types. One example of this occurrence could be the presence of long-tail relationships in a dataset where not all classes have good representation.

Next, existing literature has stressed the difficulty of using traditional evaluation techniques when using LLMs for RE (Wadhwa et al., 2023). The generative nature of the LLMs can lead to over-predictions, where the LLM output is rarely constrained to the original label space of the dataset. However, traditional forms of evaluation such as precision (P), recall (R), and f1-score (F1) require exact matching between ground truth and predicted labels (Taillé et al., 2020). Thus, this study aims to evaluate the performance of the LLMs using both traditional and modern evaluation techniques. Specifically, the evaluation protocol GenRES (Jiang et al., 2024) is employed as a modern form of evaluation. GenRES aims to evaluate the LLM predictions beyond exact matching by scoring them on qualitative aspects based on the comprehensiveness of the generated text. To this end, the following metrics were employed for this study:

- Traditional (*P*, *R*, *F1*): micro-averaged metrics used to evaluate both RC and JRE extractors.
- Topical Similarity Score (*TS*) to check if the extracted prediction closely aligns with the topic of the test samples. A higher value of *TS* indicates better topical similarity between the prediction and source text.
- Uniqueness Score (*US*) to analyze the diversity in the triples generated for JRE. A higher value of *US* indicates that the extracted triplets contain distinct and varied relationships and have low redundancy.
- Completeness Score (*CS*) to analyze how completely the extracted triples incorporate the information in the test sample. A higher value of *CS* indicates that the extracted triple successfully represents all available information.

This study concentrated on *TS*, *US*, and *CS* evaluation metrics rather than the Factualness Score (*FS*) and Granuality Score (*GS*) metrics from GenRES. This decision was made with the following rationales in mind: 1) for this study, the focus has been on the topical relevance of the extracted triples and how comprehensively they incorporate the information from the data distribution while not including redundant information. This objective can be achieved through the first three metrics. 2) While *FS* and *GS* could provide an additional layer of evaluation, they rely on LLMs for evaluation, which introduces potential instability due to their sensitivity to minor changes to prompts. Given these concerns, *FS* and *GS* metrics were not included in this study. The methodology for metric calculations can be found in Appendix A.4.

3.4 Experimental Protocol

Five popular LLMs, *"gpt-4o-mini"*, *"Meta-Llama-3.1-8B-Instruct"*, *"gemma-2-9b-it 9B"*, *"Mistral-Nemo-Instruct-2407"* and *"openchat_3.5"* were selected for this study. Their hyperparameter details can be found in Appendix A.3. The LLMs were selected to represent diverse LLM families from the literature. Next, four datasets were chosen for this work: TACRED (Zhang et al., 2017b), NYT10 (Riedel et al., 2010), FewRel (Han et al., 2018), and CrossRE (Bassignana and Plank, 2022), based on the complex data characteristics they exhibit. Details of the dataset statistics and preprocessing steps can be found in Appendix A.1. As is the trend in the literature, smaller subsets of the publicly available test sets were created by sampling 33% and 50% of the test samples for LLM-based experiments on RC and JRE, respectively. The sampling was done to conduct exhaustive experiments without surpassing the budget overhead. The sampling process employed three seed values - 13, 42, 100. Final experiments were conducted on these three sub-sampled versions of the test sets.

Next, five traditional algorithms were selected as baselines for the experiments. Algorithms for RC include RBERT (Wu and He, 2019) and LUKE (Yamada et al., 2020), PLM-based fine-tuned algorithms, and KnowPrompt (Chen et al., 2022), a prompt tuning-based algorithm. Similarly, for JRE, the algorithms SPN4RE (Sui et al., 2020) and TDEER (Li et al., 2021) were selected. The algorithms were trained in a fully supervised setting using the original training sets of the datasets. The details of each algorithm and the experimental protocol used can be found in the Appendix A.2.

Both zero-shot and few-shot experiments were conducted for the LLMs. For few-shot learning, three k values were selected - 5, 10, 20. The k-shots were retrieved using the same training set used for the traditional algorithms. Finally, ∼2100 experiments were conducted using the permutations of 4 learning strategies (*zero*, *random*, *top-k*, *vote-k*), 4 prompting strategies (*open*, *rel++*, *ent++*, *entrel++*), 3 k-shot values and 3 seed iterations. It is worth mentioning that entity type information corresponding to each entity was only present for the TACRED dataset, making it compatible with all prompting strategies. For NYT10, the entity types were extracted from the relation label, making the dataset compatible for *open* and *entrel++* for JRE and open and *rel++* for RC. Finally, since entity type information was not available for FewRel and CrossRE, the datasets were only evaluated for *open* (both RC and JRE) and *rel++* (RC) prompt types.

Finally, for evaluation, micro-precision, recall, and F1-score were used to analyze the performance of both RC and JRE models. The concept of "exact-matching" was used for the JRE models where the whole triplet was converted to a string and matched with its ground truth counterpart. Additionally, soft metrics from the GenRES benchmark discussed above were used to analyze the JRE models. Note that for the TACRED dataset, the scores depicted include the performance of the models on 'NA' class as well.

4 Results

The LLMs were evaluated with respect to different data characteristics to gauge their performance capabilities for RC and JRE. The traditional algorithms were also employed to form a basis for comparison. Based on the results obtained, it was found that the LLM-based JRE models performed extremely poorly when evaluated with the traditional metrics due to their open-ended nature. Thus, results are discussed based on the GenRES metrics for the joint extractors. Additionally, this work aims to analyze the overall efficiency of the LLMs at extracting relationships in the presence of the complex characteristics discussed above. Thus, this section examines the aggregate performance of the LLMs across multiple experiment dimensions (datasets, few-shots, seeds, learning, and prompting strategies). The details of the datasets incorporated in each complex category can be found in Table 1. More in-depth and dataset-specific results can be found in Appendix A.5.

Datasets	**fine-grained**[1]	**muliple/ overlap**[2]
NYT10	×	✓
FewRel	✓	×
CrossRE	×	✓
TACRED	✓	✓

[1] datasets with fine-grained relationships
[2] datasets with multiple relations and overlapping entities

Table 1: Dataset categorization for the study.

4.1 Fine-grained Relationships

To evaluate the performance of LLMs at extracting fine-grained relationships, the datasets were divided into **‘fine"** and **‘coarse"** categories based on the size of the label space. Specifically, TACRED and FewRel datasets were categorized as fine-grained with 42 and 80 relations, respectively. On the contrary, NYT10 and CrossRE datasets were categorized as coarse-grained, with 17 and 29 relations, respectively.

Model	Type	P	R	F1
RBERT	coarse	78.21	78.21	78.21
	fine	**88.86**	**88.86**	**88.86**
LUKE	coarse	77.35	77.35	77.35
	fine	**89.65**	**89.65**	**89.65**
KnowPrompt	coarse	75.87	75.87	75.87
	fine	**87.78**	**87.78**	**87.78**
GPT	coarse	**39.77**	**39.77**	**39.77**
	fine	9.76	9.76	9.76
Gemma	coarse	**32.95**	**32.95**	**32.95**
	fine	16.89	16.89	16.89
Llama	coarse	**25.54**	**25.54**	**25.54**
	fine	7.38	7.38	7.38
Mistral	coarse	**37.32**	**37.32**	**37.32**
	fine	8.89	8.89	8.89
Openchat	coarse	**18.63**	**18.63**	**18.63**
	fine	9.21	9.21	9.21

Table 2: Average micro-F1 of the RC models for “fine” and “coarse” dataset categories. The highest scores in each model-category pair have been highlighted. For LLM-based models, scores are averaged across 5, 10, and 20 k-shots, 4 retrieval methods, 3 seeds, and 4 prompting strategies. For PLM-based models, scores represent five cross-validation folds.

Table 2 depicts the performance of traditional and LLM-based relation classifiers. It can be observed that **LLMs were inefficient at tackling fine-grained relationships** compared to traditional algorithms like RBERT and LUKE, as can be seen by the lower performances achieved in the ‘fine“ category by the former models. Additionally, The low recall values experienced by all LLMs indicate that the models are prone to high false negatives. As the label space grows, it is common to find relationships with similar meanings. These semantically similar relations can be a probable cause of the mispredictions faced by the models. Furthermore, the performance degradation observed by the LLMs, suggests that in large label spaces, it is difficult to develop efficient prompts that can impart the full knowledge of the relation distribution.

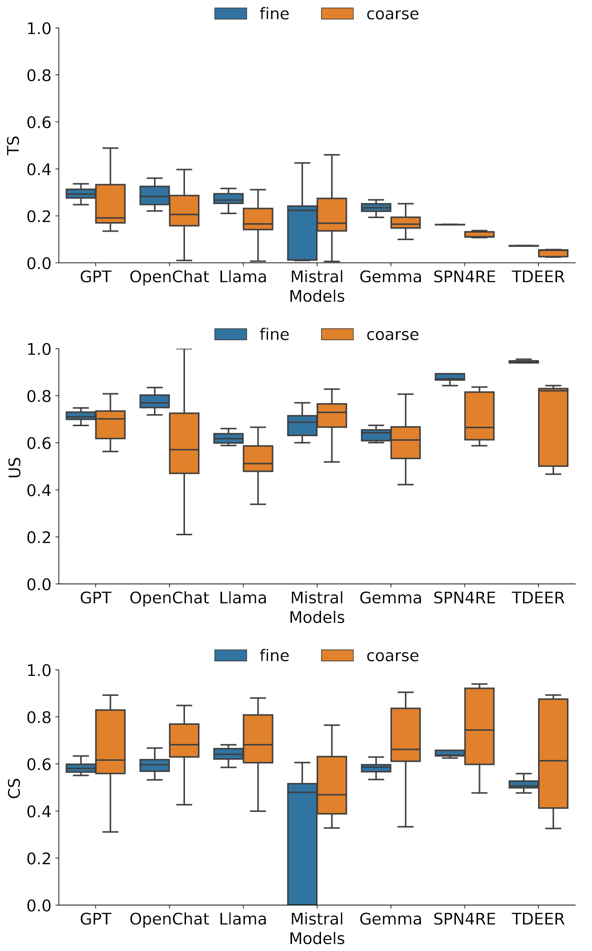


Figure 2: TS, US, and CS score distribution for “fine” and “coarse” datasets for JRE models. For LLM-based models, scores are averaged across 5, 10, and 20 k-shots, 4 retrieval methods, 3 seeds, and 2 prompting strategies. For PLM-based models, scores represent five cross-validation folds. Error bars indicate standard deviation.

Next, Figure 2 depicts the performance of traditional and LLM-based JRE models in the *TS*, *US*, and *CS* dimensions. It is apparent that the LLMs produced triples that were topically more relevant to the source text than the traditional models for fine-grained relations. It can be inferred that **in the presence of a large label space, the LLMs can better understand the distribution of the source text**. Although both categories of extractors performed comparably at extracting unique triplets, they were inefficient at extracting triplets that incorporate the complete knowledge of the source text in the presence of fine-grained relationships, as shown by the low *CS* scores in the ‘fine" category.

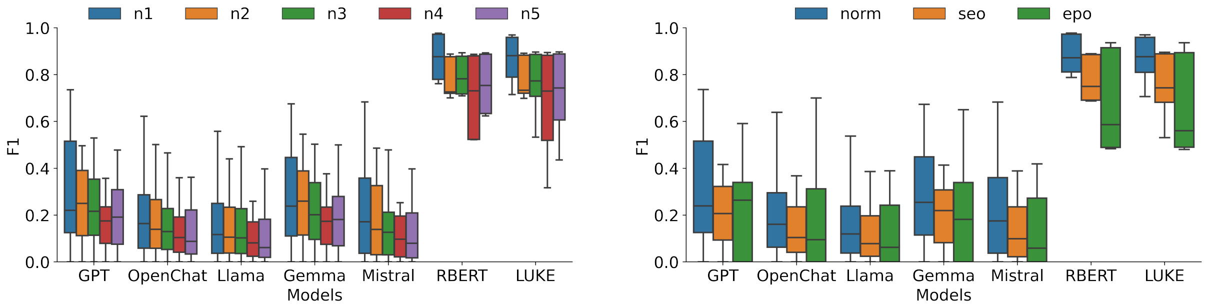


Figure 3: Micro-F1 score for RC models for multiple relations (left) and overlapping entities (right). For LLM-based models, scores are averaged across 5, 10, and 20 k-shots, 4 retrieval methods, 3 seeds, and 4 prompting strategies. For PLM-based models, scores represent five cross-validation folds. Error bars indicate standard deviation.

4.2 Multiple Relations & Overlapping Entities

This section discusses the findings conducted at the sample level where segregation was done based on the number of relations and categories of entity overlap associated with the test samples. The model performances were evaluated in the resulting subcategories. Statistics of each category can be found in Table 3 in the Appendix.

Figure 3 depicts the performance of the PLM and LLM-based RC extractors. It can be observed that, like the traditional algorithms, even **LLMs are not robust at tackling scenarios where a single or a pair of entities are shared between multiple relations.** Similarly, the performances take a hit when extracting relations from samples with more than a single relation associated with them. Such scenarios give rise to ambiguity in terms of context and relational meaning. The ambiguity, in turn, raises the possibility of multiple correct predictions, thereby confusing the extractors. Since JRE works on the principle of extracting multiple relation triplets from the same sample, it can be a possible solution to this problem.

Subsequently, the LLM-based JRE extractors were evaluated for their robustness to this issue. It was found that the LLMs were negatively impacted by this use case as **the redundancy in the triples increased and completeness decreased when multiple relations and entity overlaps were associated with the data.** This observation is depicted in Figure 4, where both the traditional and LLM-based algorithms show a gradual reduction in *US* scores across the complex categories. It can be inferred that with multiple associations, the quality of the triplets decreases, and it is more likely to have similar and redundant triplets. No topical variation was observed with this complicated scenario.

4.3 Low-resource Scenario

Finally, the performance of the LLMs was investigated for their capabilities at extracting relations in low-resource settings. Since zero-shot and few-shot experiments were conducted with the LLMs, this study investigates the variation in performance with respect to different shots of data.

Figure 5 depicts the average performance over all LLMs for RC under different demonstration retrieval strategies. Apart from a slight gain in average performance from zero to 5-shot, It can be observed that the LLMs perform consistently across the shot values. This indicates that **incorporating additional context samples does not significantly help the LLMs infer relationships**, and they are competent at tackling RC in low-resource settings.

Next, for JRE extractors, it can be inferred from Figure 6 that the LLMs can achieve better topical similarity to the source text when used for inference in zero-shot settings. This observation highlights the behavior of LLMs to concentrate on the input demonstrations (when present) rather than the target text. Furthermore, the *US* and *CS* scores stay comparable across shot values, indicating that in the presence of low resources, adding additional demonstrations does not significantly aid the extractors.

5 Discussion

This section investigates the possible reasons behind the performance degradation experienced by the LLM-based relation extractors. The influence of similar relation labels was studied for 'fine" and 'coarse" datasets for RC. Note that the LLM predictions for RC can lie in the following categories: *'correct", 'incorrect - match in the label space", 'incorrect - no match"* and *'no prediction"*. To define

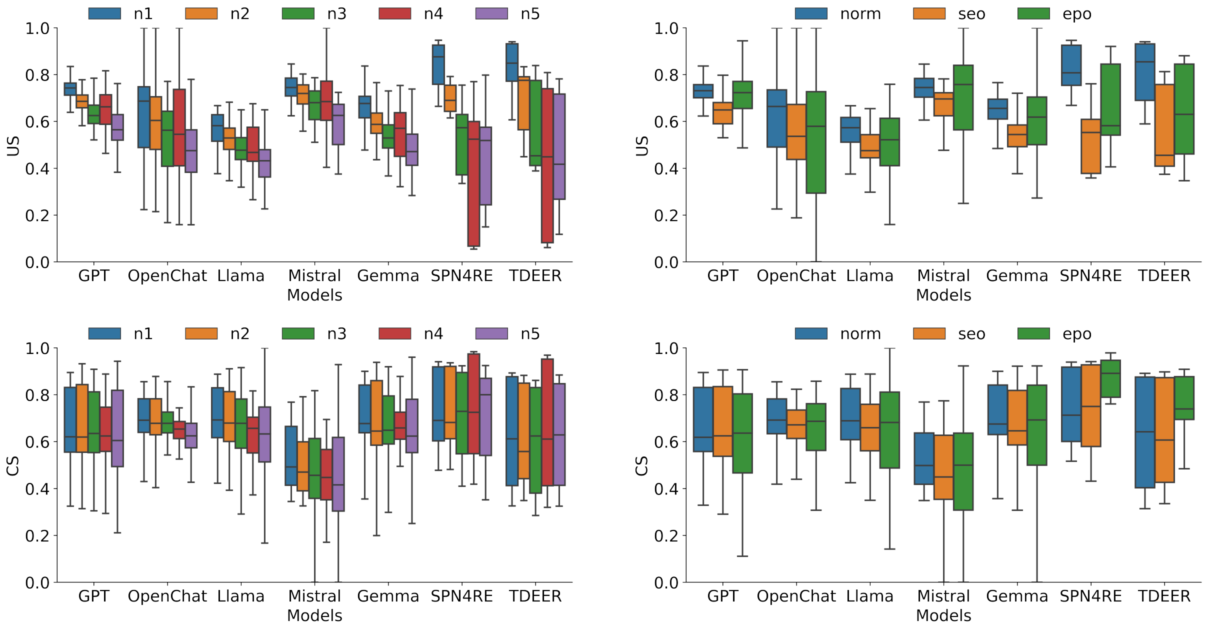


Figure 4: US and CS scores for JRE models for multiple relations (left) and overlapping entities (right). For LLM-based models, scores are averaged across 5, 10, and 20 k-shots, 4 retrieval methods, 3 seeds, and 2 prompting strategies. For PLM-based models, scores represent five cross-validation folds. Error bars indicate standard deviation.

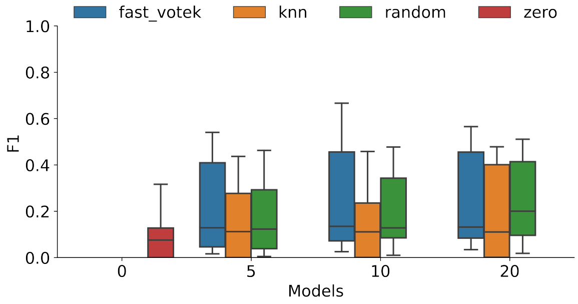


Figure 5: Micro-F1 score variation with different k-shot values for LLM-based RC models and demonstration strategies. Scores are averaged across 3 seeds, and 4 prompting strategies. Error bars indicate standard deviation.

similar relationships, the relation labels for each dataset were clustered together using word embeddings and K-means clustering. The methodology can be found in Appendix A.6. Contrary to the hypothesis presented in the previous section, it was found that in the *‘incorrect - a match in the label space"* category, the incorrect predictions did not belong to the other relations in the same cluster as the true relation. This finding suggests that the LLMs are not confused by similar relation labels.

Consequently, on investigating the source of incorrect predictions, it was found that most such instances came from the *‘incorrect - no match"* category. On calculating the cosine similarity between the *“text-embedding-ada-002"* based word embeddings of the true label and the predicted outcome, it was found that the LLMs predicted labels similar in meaning to the target label but did not exactly match the relation definition in the label space as shown in Figure 7. These false negative predictions were more apparent in the fine-grained datasets, resulting in poor performance.

Summarising the overall observations: First, as discussed above, RC models suffer from large false negative predictions, especially in the presence of large label spaces, which are common for RE. Second, JRE models were found to have an increased tendency to predict redundant triplets that lacked uniqueness when multiple relations were associated with a text sample. Finally, the incorporation of demonstrations did not significantly impact the extractors. Based on these findings, it can be inferred that similar to traditional extractors, LLM-base relation extractors are not robust to complex relation extraction scenarios. Where the traditional extractors are deterred by ambiguity caused by similar relationships, the unbounded output of the LLMs has proven to be a big obstacle to their efficient incorporation in an RE pipeline.

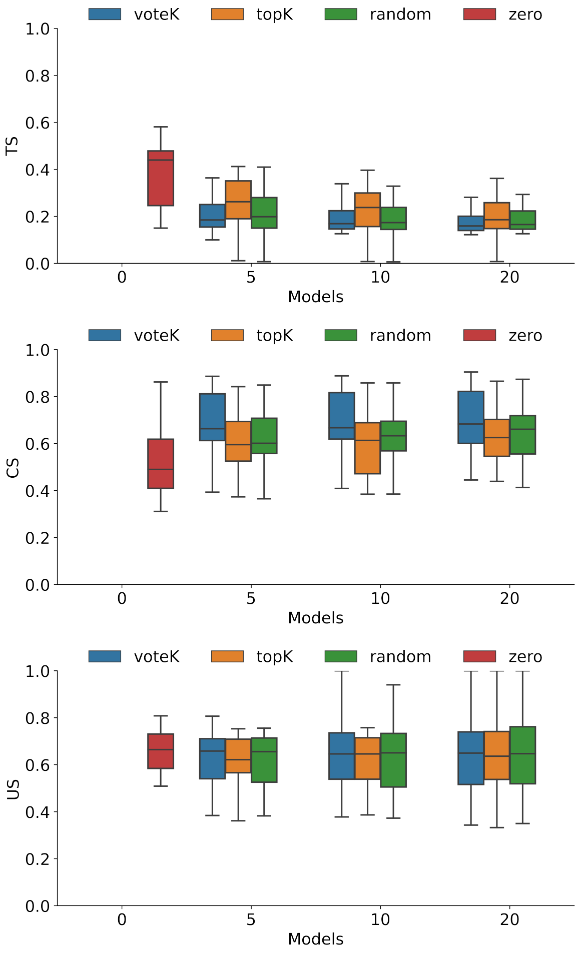


Figure 6: TS, CS and US score variations with different k-shot values. Scores are averaged across 3 seeds, and 4 prompting strategies. Error bars indicate standard deviation.

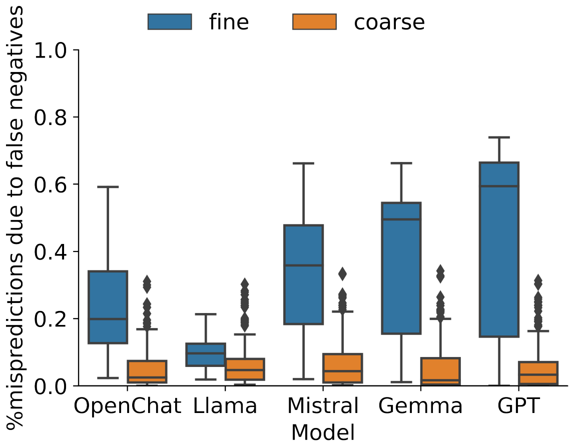


Figure 7: % incorrect predictions caused per model due to false negatives. Error bars indicate standard deviation and the dots represent outliers.

6 Conclusion

This study explores the capabilities of five state-of-the-art LLMs to extract relationships from textual data in the presence of complex data characteristics. To facilitate this, both RC and JRE paradigms in RE were investigated using a comprehensive set of experiments with multiple LLMs, demonstration retrieval, and prompting strategies against five traditional RE algorithms. The study raised concerns regarding the feasibility of LLMs for RE. The low performances combined with the LLMs' brittleness to the complex attributes act as a call for action to better understand the LLM's behavior with future work.

7 Limitations

This feasibility study has a few potential limitations related to the protocols used for the experiments. First, the datasets selected for this work were constrained at the sentence level. The performance of LLMs for extracting relations at the document level, which is a more challenging use case, was not explored in this study. Second, test-retest experiments that test the stability of the LLMs predictions were not incorporated into the study owing to the large size of the experimentation protocol. Finally, this work was constrained to studying the extracted triplets' topical consistency, uniqueness, and completeness. Future work incorporating factual and granular metrics could help paint a more precise picture of the JRE models.

References

Mehmet Aydar, Ozge Bozal, and Furkan Ozbay. 2020. Neural relation extraction: a survey. *arXiv preprint arXiv:2007.04247*.

Elisa Bassignana and Barbara Plank. 2022. What do you mean by relation extraction? a survey on datasets and study on scientific relation classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 67–83, Dublin, Ireland. Association for Computational Linguistics.

Tom B Brown. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Xiang Chen, Ningyu Zhang, Xin Xie, Shumin Deng, Yunzhi Yao, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. 2022. Knowprompt: Knowledge-aware prompt-tuning with synergistic optimization for relation extraction. In *Proceedings of the ACM Web conference 2022*, pages 2778–2788.

Jacob Devlin. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. 2022. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*.

Jiale Han, Shuai Zhao, Bo Cheng, Shengkun Ma, and Wei Lu. 2022. Generative prompt tuning for relation classification. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3170–3185, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Xu Han, Tianyu Gao, Yankai Lin, Hao Peng, Yaoliang Yang, Chaojun Xiao, Zhiyuan Liu, Peng Li, Jie Zhou, and Maosong Sun. 2020. More data, more relations, more context and more openness: A review and outlook for relation extraction. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 745–758, Suzhou, China. Association for Computational Linguistics.

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. *Preprint*, arXiv:1810.10147.

Pengcheng Jiang, Jiacheng Lin, Zifeng Wang, Jimeng Sun, and Jiawei Han. 2024. Genres: Rethinking evaluation for generative relation extraction in the era of large language models. *arXiv preprint arXiv:2402.10744*.

Joohong Lee, Sangwoo Seo, and Yong Suk Choi. 2019. Semantic relation classification via bidirectional lstm networks with entity-aware attention using latent entity typing. *Preprint*, arXiv:1901.08163.

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

Xianming Li, Xiaotian Luo, Chenghao Dong, Daichuan Yang, Beidi Luan, and Zhen He. 2021. TDEER: An efficient translating decoding schema for joint extraction of entities and relations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8055–8064, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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

Yinhan Liu. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

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

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.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Machine Learning and Knowledge Discovery in Databases*, pages 148–163, Berlin, Germany. Springer Berlin Heidelberg.

Hongjin Su, Jungo Kasai, Chen Henry Wu, Weijia Shi, Tianlu Wang, Jiayi Xin, Rui Zhang, Mari Ostendorf, Luke Zettlemoyer, Noah A Smith, et al. 2022. Selective annotation makes language models better few-shot learners. *arXiv preprint arXiv:2209.01975*.

Dianbo Sui, Yubo Chen, Kang Liu, Jun Zhao, Xiangrong Zeng, and Shengping Liu. 2020. Joint entity and relation extraction with set prediction networks. *Preprint*, arXiv:2011.01675.

Anushka Swarup, Avanti Bhandarkar, Olivia P. Dizon-Paradis, Ronald Wilson, and Damon L. Woodard. 2024. Maximizing relation extraction potential: A data-centric study to unveil challenges and opportunities. *IEEE Access*, 12:167655–167682.

Bruno Taillé, Vincent Guigue, Geoffrey Scoutheeten, and Patrick Gallinari. 2020. Let's Stop Incorrect Comparisons in End-to-end Relation Extraction! In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3689–3701, Online. Association for Computational Linguistics.

Ryuichi Takanobu, Tianyang Zhang, Jiexi Liu, and Minlie Huang. 2019. A hierarchical framework for relation extraction with reinforcement learning. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'19/IAAI'19/EAAI'19. AAAI Press.

Wei Tang, Benfeng Xu, Yuyue Zhao, Zhendong Mao, Yifeng Liu, Yong Liao, and Haiyong Xie. 2022. UniRel: Unified representation and interaction for joint relational triple extraction. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7087–7099, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay

Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Somin Wadhwa, Silvio Amir, and Byron 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)*, pages 15566–15589, Toronto, Canada. Association for Computational Linguistics.

Zhen Wan, Fei Cheng, Zhuoyuan Mao, Qianying Liu, Haiyue Song, Jiwei Li, and Sadao Kurohashi. 2023. Gpt-re: In-context learning for relation extraction using large language models. *arXiv preprint arXiv:2305.02105*.

Yucheng Wang, Bowen Yu, Yueyang Zhang, Tingwen Liu, Hongsong Zhu, and Limin Sun. 2020. TPLinker: Single-stage joint extraction of entities and relations through token pair linking. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1572–1582, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.

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

Shanchan Wu and Yifan He. 2019. Enriching pre-trained language model with entity information for relation classification. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM '19, page 2361–2364, New York, NY, USA. Association for Computing Machinery.

Xin Xu, Yuqi Zhu, Xiaohan Wang, and Ningyu Zhang. 2023. How to unleash the power of large language models for few-shot relation extraction? *arXiv preprint arXiv:2305.01555*.

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. LUKE: Deep contextualized entity representations with entity-aware self-attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6442–6454, Online. Association for Computational Linguistics.

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

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

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451, Florence, Italy. Association for Computational Linguistics.

Kang Zhao, Hua Xu, Yue Cheng, Xiaoteng Li, and Kai Gao. 2021. Representation iterative fusion based on heterogeneous graph neural network for joint entity and relation extraction. *Knowledge-Based Systems*, 219:106888.

Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 207–212, Berlin, Germany. Association for Computational Linguistics.

Wenxuan Zhou and Muhao Chen. 2022. An improved baseline for sentence-level relation extraction. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 161–168, Online only. Association for Computational Linguistics.

A Appendix

A.1 Datasets

The datasets used for this study have been discussed below, and their statistics can be found in Tables 3.

NYT10 (Riedel et al., 2010): A popular RE dataset created by aligning relations from the Freebase knowledge base with the New York Times (NYT) corpus. This study uses the preprocessed version of the dataset by Takanobu et al. (2019). The dataset exhibits complex issues such as multiple relations, overlapping entities, and long-tail distribution.

TACRED (Zhang et al., 2017b): A large-scale RE dataset built using newswire and web text. The dataset contains relations used in the TAC KBP challenges. The dataset exhibits complex characteristics such as fine-grained relations, multiple

Dataset	#rels	#test	seed	norm	seo	epo	n1	n2	n3	n4	n5
CrossRE	17	3026	13	130	2339	557	76	171	251	328	2200
		3026	100	131	2308	587	84	169	233	364	2176
		3026	42	146	2282	598	86	187	235	337	2181
NYT10	29	1934	13	990	228	716	982	391	184	304	73
		1934	100	964	214	756	957	389	193	328	67
		1934	42	1008	219	707	998	377	189	302	68
TACRED	42	5118	13	1134	3004	980	1117	776	626	506	2093
		5118	100	1086	3081	951	1075	824	604	530	2085
		5118	42	1112	3035	971	1103	757	629	517	2112
FewRel	80	2412	13	2394	18	0	2391	21	0	0	0
		2412	100	2380	31	1	2379	33	0	0	0
		2412	42	2395	16	1	2391	21	0	0	0

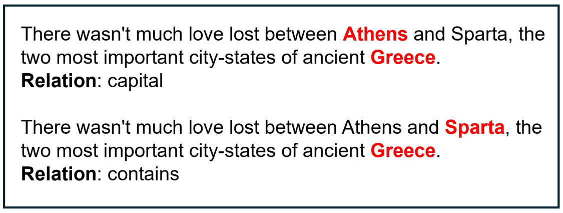
Table 3: Test Data Statistics depicting the number of samples in each category as per the JRE paradigm. *#rels*: number of relations; *#test*: number of test samples; *norm*: normal samples (without overlapping entities); *seo*: single entity overlap; *epo*: entity pair overlap; *n1*: one relation per sample; *n2*: two relations per sample; *n3*: three relations per sample; *n4*: four relations per sample; *n5*: five or more relations per sample.

relations, overlapping entities, and long-tail distribution.

FewRel (Han et al., 2018): A large-scale few-shot learning dataset with a balanced relation distribution. The preprocessed version of the dataset by Zhang et al. (2019) was used where the predefined train and validation sets were mixed, and one hundred instances from each class were sampled for the training set and 200 for the validation and test set. The resulting dataset had 27,328 samples with 80 relation classes. The dataset exhibits fine-grained relationships.

CrossRE (Bassignana and Plank, 2022): A multi-domain RE dataset that contains data from six domains. For this study, data from all domains was combined to create the train and test set. The dataset exhibits complex characteristics such as multiple relations, overlapping entities, and long-tail distribution.

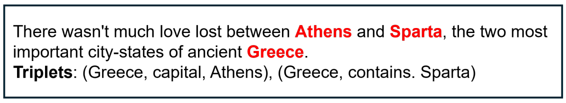
Most datasets mentioned above were originally released for the RC task with entity mentions annotated with each text sample. To make them compatible with the JRE setting, this work grouped common text samples together and created triplets using the existing entity and relation annotations. Figures 8 and 9 show an example of a text sample under the RC and JRE settings.



There wasn't much love lost between **Athens** and Sparta, the two most important city-states of ancient **Greece**.
Relation: capital

There wasn't much love lost between Athens and **Sparta**, the two most important city-states of ancient **Greece**.
Relation: contains

Figure 8: Relation Classification: the same sentence acts as two text samples depending on the target entities.



There wasn't much love lost between **Athens** and **Sparta**, the two most important city-states of ancient **Greece**.
Triplets: (Greece, capital, Athens), (Greece, contains. Sparta)

Figure 9: Joint Relation Extraction: a text sample with multiple relations.

A.2 Traditional Algorithms

The following five PLM-based traditional algorithms were employed as baselines for this study:

RBERT: uses a BERT (Devlin, 2018) encoder to extract contextual representations of the text sample. The information about the position of entities is added using special tokens.

LUKE: uses a pre-trained RoBERTa (Liu, 2019) encoder, which is made to learn the representation of entities as separate tokens along with the text sample.

KnowPrompt: incorporates information about the label space by adding answer words in input prompts. The representation of these words is optimized using a RoBERTa encoder.

SPN4RE: uses a bidirectional decoder to extract entity relation triplets. The algorithm introduces a bipartite loss function to circumvent the ordered extraction of triplets.

TDEER: uses a decomportion-based framework where first, all entity tokens are extracted using a binary classifier. This stage is followed by relation extraction using a multi-label classifier.

These algorithms were trained on the selected datasets using a fully supervised protocol. Five-fold cross-validation was performed during training. The trained models were tested on the three seed subsets of test data. The hyperparameters from the original implementations of the algorithms were used.

A.3 LLM Hyperparameters

Open-source models, including *"Meta-Llama-3.1-8B-Instruct"*, *"gemma-2-9b-it 9B"*, *"Mistral-Nemo-Instruct-2407"*, and *"openchat_3.5"*, were utilized via the Hugging Face library[2]. The *max_new_tokens* parameter was set to 300 tokens for JRE and 128 tokens for RC, while all other hyperparameters were left at their default settings. Similarly, for OpenAI's *"gpt-4o-mini"*, the *max_tokens* parameter was configured to 300 for JRE and 128 for RC, with a *temperature* of 0 and *top_p* set to 1.

A.4 GenRES Methodology

For calculating the GenRES metrics, the methodology followed by Jiang et al. (2024) was used. LDA-based topic models were calculated for TS calculation using the test datasets. Number of topics for all datasets was set to 150. For CS and US calculations, OpenAI's *"text-embedding-ada-002"* embeddings were incorporated.

A.5 Results

Tables 6 and 7 depict the dataset level scores for the JRE models used in this study. Similarly, Tables 4 and 5 details similar scores for RC models. The reported scores have been averaged across the k-shots used for this study owing to the observation that significant performance gain was not observed with the models on using greater shots of data.

[2] https://huggingface.co/

Datasets	**Models**	**Demo**	**P**	**R**	**F1**
FewRel	GPT	voteK	41.61	41.61	41.61
		topK	25.91	25.91	25.91
		random	27.9	27.9	27.9
		zero	16.54	16.54	16.54
	Gemma	voteK	41.79	41.79	41.79
		topK	32.75	32.75	32.75
		random	30.09	30.09	30.09
		zero	17.22	17.22	17.22
	Llama	voteK	34.84	34.84	34.84
		topK	23.46	23.46	23.46
		random	25.06	25.06	25.06
		zero	15.84	15.84	15.84
	Mistral	voteK	33.73	33.73	33.73
		topK	23.68	23.68	23.68
		random	25.07	25.07	25.07
		zero	11.92	11.92	11.92
	OpenChat	voteK	38.93	38.93	38.93
		topK	24.81	24.81	24.81
		random	25.57	25.57	25.57
		zero	14.68	14.68	14.68

Continued on next page

Table 4 Continued from previous page

Datasets	Models	Demo	P	R	F1
NYT10	GPT	voteK	50.23	50.23	50.23
		topK	39.87	39.87	39.87
		random	41.05	41.05	41.05
		zero	25.35	25.35	25.35
	Gemma	voteK	47.34	47.34	47.34
		topK	38.53	38.53	38.53
		random	39.65	39.65	39.65
		zero	21.39	21.39	21.39
	Llama	voteK	47.35	47.35	47.35
		topK	36.96	36.96	36.96
		random	38.91	38.91	38.91
		zero	21.46	21.46	21.46
	Mistral	voteK	46.65	46.65	46.65
		topK	33.49	33.49	33.49
		random	36.55	36.55	36.55
		zero	23.1	23.1	23.1
	OpenChat	voteK	46.54	46.54	46.54
		topK	33.09	33.09	33.09
		random	37.3	37.3	37.3
		zero	20.32	20.32	20.32
CrossRE	GPT	voteK	30.97	30.97	30.97
		topK	17.89	17.89	17.89
		random	24.05	24.05	24.05
		zero	9.79	9.79	9.79
	Gemma	voteK	32.65	32.65	32.65
		topK	16.78	16.78	16.78
		random	23.05	23.05	23.05
		zero	11.19	11.19	11.19
	Llama	voteK	16.14	16.14	16.14
		topK	9.12	9.12	9.12
		random	12.9	12.9	12.9
		zero	5.13	5.13	5.13
	OpenChat	voteK	26.17	26.17	26.17
		topK	10.76	10.76	10.76
		random	18.42	18.42	18.42
		zero	5.09	5.09	5.09

Continued on next page

Table 4 Continued from previous page

Datasets	Models	Demo	P	R	F1
TACRED	GPT	voteK	9.1	9.1	9.1
		topK	5.71	5.71	5.71
		random	7.35	7.35	7.35
		zero	5.42	5.42	5.42
	Gemma	voteK	9.79	9.79	9.79
		topK	5.6	5.6	5.6
		random	8.03	8.03	8.03
		zero	5.79	5.79	5.79
	Llama	voteK	7.38	7.38	7.38
		topK	2.11	2.11	2.11
		random	5.56	5.56	5.56
		zero	4.63	4.63	4.63
	Mistral	voteK	7.97	7.97	7.97
		topK	5.57	5.57	5.57
		random	6.92	6.92	6.92
		zero	6.26	6.26	6.26
	OpenChat	voteK	9.55	9.55	9.55
		topK	4.05	4.05	4.05
		random	7.4	7.4	7.4
		zero	5.35	5.35	5.35

Table 4: Traditional metrics for RC LLM-based models across 0, 5, 10, and 20 few-shot, 4 prompting and 3 seed strategies.

Dataset	**Model**	**P**	**R**	**F1**
FewRel	RBERT	89.31	89.31	89.31
	LUKE	90.55	90.55	90.55
	KnowPrompt	88.38	88.38	88.38
NYT10	RBERT	81.36	81.36	81.36
	LUKE	80.41	80.41	80.41
	KnowPrompt	76.9	76.9	76.9
CrossRE	RBERT	75.06	75.06	75.06
	LUKE	74.29	74.29	74.29
	KnowPrompt	74.84	74.84	74.84
TACRED	RBERT	88.42	88.42	88.42
	LUKE	88.74	88.74	88.74
	KnowPrompt	87.17	87.17	87.17

Table 5: Traditional metrics for RC PLM-based models across five cross-validation folds.

Datasets	**Models**	**Demo**	**P**	**R**	**F1**	**TS**	**CS**	**US**
FewRel	GPT	voteK	5.74	9.15	6.62	26.58	60.11	72.43
		topK	1.19	2.2	1.44	31.56	56.53	69.79
		random	1.76	3.3	2.14	29.16	57.16	72.91
		zero	0.25	0.66	0.35	49.6	62.68	68.15
	Gemma	voteK	3.92	5.81	4.42	12.5	33.79	36.75
		topK	2.03	3.57	2.42	25.03	57.67	63.28
		random	2.77	4.68	3.25	22.58	56.07	63.07
		zero	0.2	0.44	0.26	42.99	60.29	60.33
	Llama	voteK	3.9	7.4	4.68	24.82	66.95	63.28
		topK	0.93	2.13	1.19	27.73	58.6	62.89
		random	1.29	2.98	1.66	27.59	65.1	61.49
		zero	0.09	0.3	0.12	47.94	62.02	53.79
	Mistral	voteK	2.58	3.86	2.92	12.85	29.43	69.58
		topK	0.41	0.75	0.49	12.14	21.65	63.76
		random	1.68	2.82	1.97	23.07	46.79	72.64
		zero	0.16	0.54	0.24	42.25	59.43	61.52
	OpenChat	voteK	7.39	10.82	8.14	25.42	62.55	79.54
		topK	1.97	3.71	2.35	30.49	57.44	75.42
		random	2.6	4.43	2.99	28.23	57.23	78.88
		zero	0.33	1.24	0.5	57.81	64.06	66.51

Continued on next page

Table 6 Continued from previous page

Datasets	Models	Demo	P	R	F1	TS	CS	US
NYT10	GPT	voteK	25.44	30.46	25.54	16.05	86.95	73.97
		topK	16.47	19.41	16.33	18.04	80.57	72
		random	14.32	17.19	14.27	16.78	82.51	73.72
		zero	5.17	8.13	5.78	32.37	65.07	73
	Gemma	voteK	28.42	34.45	28.72	14.54	88.06	65.76
		topK	18.82	23.01	19.06	15.64	83.82	63.08
		random	11.57	14.22	11.68	12.51	65.89	72.85
		zero	4.81	9.16	5.83	27.6	65.48	64.31
	Llama	voteK	22.99	32.31	23.68	14.58	86.15	59.1
		topK	7.57	11.48	7.98	10.39	46.69	73.4
		random	11.23	16.95	11.63	10.63	61.5	42.68
		zero	1.28	3.67	1.58	35.06	60.5	60.68
	Mistral	voteK	22.69	24.6	21.89	14.63	72.99	78.38
		topK	1.26	1.5	1.24	3.06	7.73	96.79
		random	13.25	14.64	12.72	15.43	66.6	75.7
		zero	3.1	7.93	3.84	32.72	53.22	68.76
	OpenChat	voteK	22.3	27.43	22.23	14.69	82.76	72.35
		topK	12.61	15.92	12.44	15.84	77.35	68.82
		random	11.36	13.77	11	14.56	71.78	76.95
		zero	2.21	7.32	2.97	35.86	61.38	65.57
CrossRE	GPT	voteK	16.45	15.52	15.26	33.22	66.03	57.66
		topK	4.53	4.74	4.28	38.68	60.65	60.53
		random	6.68	6.62	6.28	38.25	59.28	59.16
		zero	0.08	0.14	0.09	48.42	53.7	60.69
	Gemma	voteK	22.07	19.31	19.67	22.21	69.63	47.21
		topK	0.93	0.98	0.91	9.72	18	85.32
		random	10.73	9.86	9.76	25.36	64.53	48.95
		zero	0.02	0.09	0.04	44.66	52.02	52.98
	Llama	voteK	17.46	18.89	17.26	23.74	72.82	48.41
		topK	5.53	7.13	5.82	26.5	66.8	49.97
		random	8.86	10.68	9.1	26.68	68.15	50.56
		zero	0.08	0.11	0.09	50.79	53.76	51.4
	Mistral	voteK	14.53	13.72	13.46	28.64	62.31	52.47
		topK	3.23	3.75	3.19	33.56	58.25	54.33
		random	0.48	0.5	0.46	7.6	12.1	11.64
		zero	0.03	0.11	0.05	45.41	51.26	54.3
	OpenChat	voteK	18.61	17.44	17.16	28.39	69.54	54.17
		topK	4.62	4.88	4.43	35.04	61.95	57.84
		random	7.2	7.46	6.86	32.69	63.83	55.4
		zero	0.14	0.19	0.16	52.37	57.19	60.01
	GPT	voteK	3.26	4.06	3.18	19.12	59.21	65.89
		topK	1.67	2.21	1.67	25.97	52.24	66.07
		random	2.6	3.13	2.5	19.56	53.92	69.56
		zero	1.57	2.13	1.61	33.03	33.96	79.95

Continued on next page

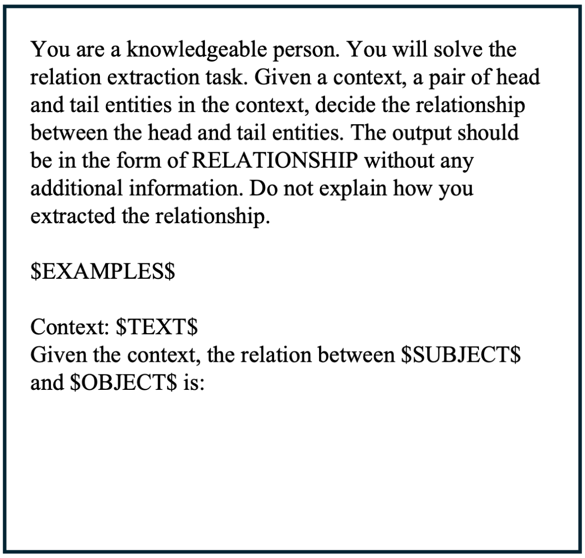
Table 6 Continued from previous page

Datasets	**Models**	**Demo**	**P**	**R**	**F1**	**TS**	**CS**	**US**
	Gemma	voteK	1.97	2.24	1.82	15.94	63.89	58.08
		topK	0.7	0.92	0.7	21.42	61.72	52.31
		random	1.47	1.68	1.37	16.42	62.32	59.05
		zero	1.26	1.8	1.3	26.35	38.91	71.84
	Llama	voteK	1.4	2.24	1.45	11.79	46.51	63.65
		topK	0.27	0.44	0.28	22.35	64.72	44.87
		random	1.3	2.04	1.33	16.47	64.36	44.98
		zero	0.23	0.65	0.27	34.55	42.14	61.01
	Mistral	voteK	3.21	3.53	2.91	15.85	46.7	70.6
		topK	0.74	1.22	0.79	29.6	41.9	71.5
		random	2.24	2.44	2.01	16.43	42.62	71.16
		zero	0.97	1.85	1.08	34.85	37.14	74.24
	OpenChat	voteK	2.1	3.15	2.12	19.37	65.69	43.07
		topK	0.29	0.64	0.34	28.01	65.86	29.56
		random	1.23	1.93	1.26	19.96	61.97	44.09
		zero	0.38	0.94	0.46	41.49	41.14	72

Table 6: Traditional and GenRES metrics for JRE LLM-based models across 0, 5, 10, and 20 few-shot, 4 prompting and 3 seed strategies.

Dataset	Model	P	R	F1	TS	CS	US
FewRel	SPN4RE	35.42	39.76	36.81	16.25	65.06	89.02
	TDEER	3.87	4.04	3.92	7.27	51.36	94.48
NYT10	SPN4RE	74.88	68.67	70.22	10.99	92.85	82.54
	TDEER	29.22	24.37	26	5.53	88.29	82.81
CrossRE	SPN4RE	35.35	30.86	31.44	13.5	74.25	60.12
	TDEER	5.07	5.12	4.83	2.58	60.6	48.77
TACRED	SPN4RE	26.38	28.15	24.45	11.12	55.29	68.57
	TDEER	18.43	16.88	15.99	5.39	36.9	82.02

Table 7: Traditional and GenRES metrics for JRE PLM-based models across five cross-validation folds.



You are a knowledgeable person. You will solve the relation extraction task. Given a context, a pair of head and tail entities in the context, decide the relationship between the head and tail entities. The output should be in the form of RELATIONSHIP without any additional information. Do not explain how you extracted the relationship.

$EXAMPLES$

Context: $TEXT$
Given the context, the relation between $SUBJECT$ and $OBJECT$ is:

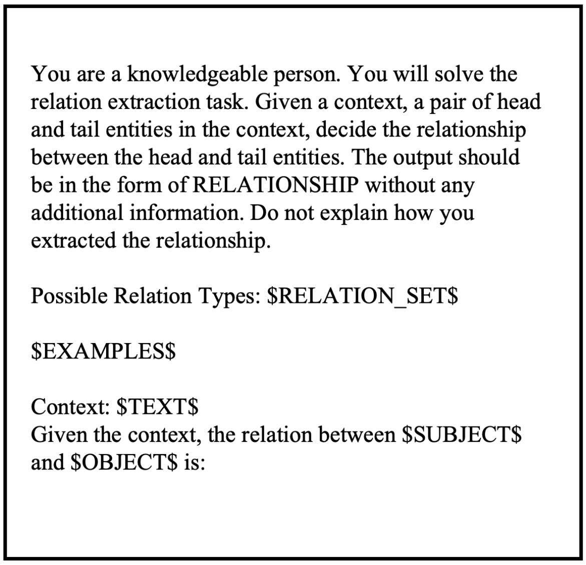
Figure 10: Prompt RC: **open**

A.6 Clustering Methodology

To investigate the influence of similar relation labels on the extraction process, the label space was clustered using word embeddings of the label verbalizations. The word embeddings were calculated using the *“text-embedding-ada-002"* embeddings, and K-means was used for clustering. Four sets of clustering experiments were conducted for each dataset. The K values were set to (3, 4, 5, 6) for NYT10, TACRED, and CrossRE and (9, 10, 11, 12) for FewRel. The values were chosen through manual optimization.

A.7 RC and JRE Prompt Templates

The prompts used for RC are shown in Figures 10, 11, 13, and 12. Similarly, Figures 14 and 15 were used for JRE.



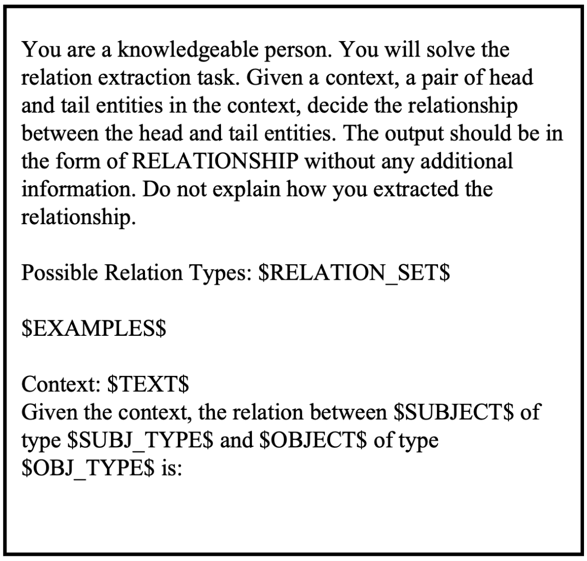
You are a knowledgeable person. You will solve the relation extraction task. Given a context, a pair of head and tail entities in the context, decide the relationship between the head and tail entities. The output should be in the form of RELATIONSHIP without any additional information. Do not explain how you extracted the relationship.

Possible Relation Types: $RELATION_SET$

$EXAMPLES$

Context: $TEXT$
Given the context, the relation between $SUBJECT$ and $OBJECT$ is:

Figure 11: Prompt RC: **rel++**



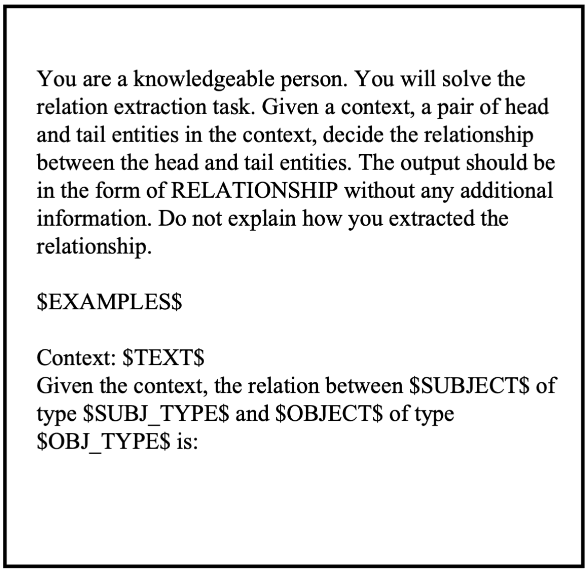
You are a knowledgeable person. You will solve the relation extraction task. Given a context, a pair of head and tail entities in the context, decide the relationship between the head and tail entities. The output should be in the form of RELATIONSHIP without any additional information. Do not explain how you extracted the relationship.

Possible Relation Types: $RELATION_SET$

$EXAMPLES$

Context: $TEXT$
Given the context, the relation between $SUBJECT$ of type $SUBJ_TYPE$ and $OBJECT$ of type OBJ_TYPE is:

Figure 12: Prompt RC: **entrel++**

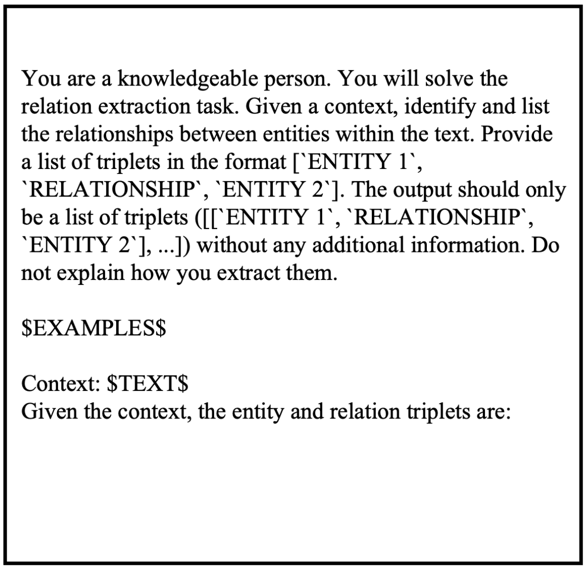


You are a knowledgeable person. You will solve the relation extraction task. Given a context, a pair of head and tail entities in the context, decide the relationship between the head and tail entities. The output should be in the form of RELATIONSHIP without any additional information. Do not explain how you extracted the relationship.

$EXAMPLES$

Context: $TEXT$
Given the context, the relation between $SUBJECT$ of type $SUBJ_TYPE$ and $OBJECT$ of type OBJ_TYPE is:

Figure 13: Prompt RC: **ent++**

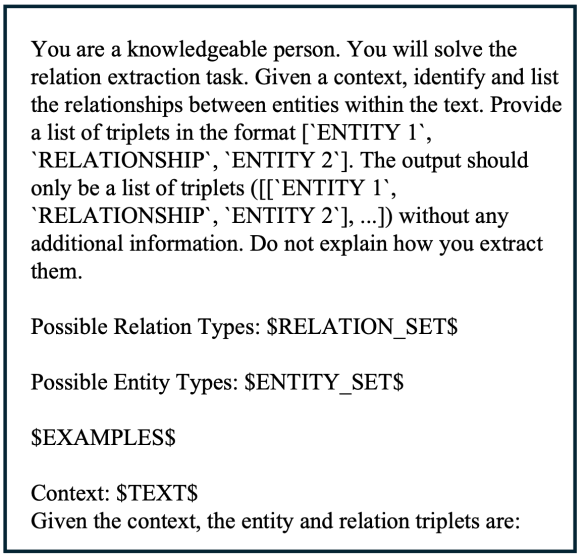


You are a knowledgeable person. You will solve the relation extraction task. Given a context, identify and list the relationships between entities within the text. Provide a list of triplets in the format [`ENTITY 1`, `RELATIONSHIP`, `ENTITY 2`]. The output should only be a list of triplets ([[`ENTITY 1`, `RELATIONSHIP`, `ENTITY 2`], ...]) without any additional information. Do not explain how you extract them.

$EXAMPLES$

Context: $TEXT$
Given the context, the entity and relation triplets are:

Figure 14: Prompt JRE: **open++**



You are a knowledgeable person. You will solve the relation extraction task. Given a context, identify and list the relationships between entities within the text. Provide a list of triplets in the format [`ENTITY 1`, `RELATIONSHIP`, `ENTITY 2`]. The output should only be a list of triplets ([[`ENTITY 1`, `RELATIONSHIP`, `ENTITY 2`], ...]) without any additional information. Do not explain how you extract them.

Possible Relation Types: $RELATION_SET$

Possible Entity Types: $ENTITY_SET$

$EXAMPLES$

Context: $TEXT$
Given the context, the entity and relation triplets are:

Figure 15: Prompt JRE: **entrel++**