

Retrieval-Augmented Generation-based Relation Extraction

Sefika Efeoglu^{a,*} and Adrian Paschke^{a,b}

^a *Department of Computer Science, Freie Universitaet Berlin, Berlin, Germany*

E-mail: adrian.paschke@fokus.fraunhofer.de

^b *Data Analytic Center (DANA), Fraunhofer Institute FOKUS, Berlin, Germany*

Abstract.

Information Extraction (IE) is a transformative process that converts unstructured text data into a structured format by employing entity and relation extraction (RE) methodologies. Identifying the relation between a pair of entities plays a crucial role within this framework. Despite the availability of various techniques for RE, their efficacy heavily depends on access to labeled data and substantial computational resources. To address these challenges, Large Language Models (LLMs) have emerged as promising solutions; however, they are prone to generating hallucinated responses due to the limitations of their training data. To overcome these shortcomings, this work proposes a Retrieval-Augmented Generation-based Relation Extraction (RAG4RE) approach to enhance RE performance.

We evaluate the effectiveness of RAG4RE using various LLMs. By leveraging established benchmarks such as TACRED, TACREV, Re-TACRED and SemEval RE datasets, we aim to comprehensively assess the efficacy of our methodology. Specifically, we employ prominent LLMs, including Flan T5, Llama2, and Mistral, in our investigation. The results of our work demonstrate that RAG4RE outperforms traditional RE methods based solely on LLMs, with significant improvements observed in the TACRED dataset and its variations. Furthermore, our approach exhibits remarkable performance compared to previous RE methodologies across both TACRED and TACREV datasets, underscoring its efficacy and potential for advancing RE tasks in natural language processing.

Keywords: Relation Extraction, Large Language Models, RAG

1. Introduction

Information Extraction (IE) is a process of converting unstructured text data into structured data by applying entity and relation extraction approaches. Identifying the relation between a pair of entities in a sentence, Relation Extraction (RE), is one of the most significant tasks in the IE pipeline [1]. RE plays a pivotal role in constructing domain-specific Knowledge Graphs (KGs) from text data and ensuring the completeness of KGs. An example of a relation type between entity pairs, such as *per:cities_of_residence*, is illustrated in Figure 1, where the head entity (*Eugenio Vagni*) is linked to the tail entity (*Sulu*). Various RE approaches have been developed, including supervised RE, distant supervision, unsupervised RE methods, rule-based and semi-supervised approaches (e.g., weakly supervised RE) [2–6]. However, well-performing RE approaches, e.g., supervised learning, require a large amount of labeled data and substantial computation time. Another effective method for identifying relation types between entities is fine-tuning language models [7–12]. It is important to note, however, that both supervised learning approaches and fine-tuning language models demand significant GPU memory and computational time during their training phase.

*Corresponding author. E-mail: sefika.efeoğlu@fu-berlin.de.

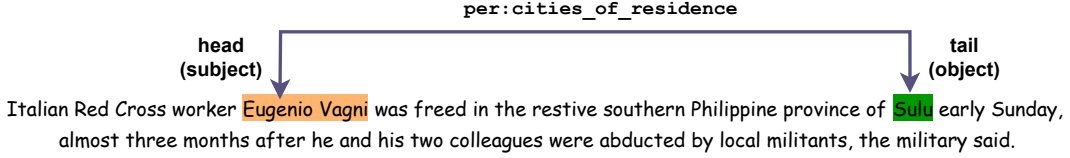


Fig. 1. Example of a relation between head and tail entities in a sentence.

General-purpose Large Language Models (LLMs) exhibit remarkable inference capabilities when applied with zero-shot prompting techniques, allowing them to effectively handle key tasks in IE, such as Entity Recognition (ER) and RE. However, they are prone to generating hallucinated outputs when lacking prior knowledge, owing to the next-token prediction mechanism inherent in these autoregressive models. Additionally, LLM prompt-tuning approaches require both prompt template engineering and domain experts for domain-specific IE approaches [8]. However, template engineering is time-consuming due to its manual nature. Retrieval-Augmented Generation (RAG) has been proposed to reduce hallucinations in LLMs when LLM-based conversational systems produce random responses to queries [13]. The RAG system functions akin to an open-book exam, integrating relevant information from the Embedding Database directly into the query (sentence) [13]. Although there have been attempts to apply the LLM approach in conjunction with zero-shot prompting techniques, such as multiple-choice questioning [14] and rationale prompting [15], these works underperform on RE benchmarks due to a high number of false predictions. Specifically, they do not incorporate relevant context or information into the query sentence within the prompt template. As a result, the RAG, using zero-shot settings, could improve and reduce false predictions of LLMs in identifying relation types between entity pairs in sentences.

Well-performing RE approaches, e.g., supervised learning, require a large amount of labeled training data and significant computational time, as they learn RE patterns in a supervised manner. Another effective method—fine-tuning language models—requires considerable GPU memory and computational time, particularly when both the base model size and training data are large, as the base model weights and training data must be loaded into the GPU to facilitate efficient training [12]. In the era of LLMs, well-designed prompts (or the prompt engineering approaches) might help us identify the relations between entities in a sentence. It is clear that well-designed prompts yield highly accurate performance in other downstream tasks, e.g., ontology-driven knowledge graph generation [16], text-to-image generation [17] by including information about fictional characters in the prompt template as an example of zero-shot settings, and ontology matching [18] by including information about the concept to be matched in the prompts. Building on previous works that utilize zero-shot prompting, enriching the context of the prompt could provide task-relevant information to the LLMs, improving their responses. This can be done while still preserving the zero-shot settings of the prompt within the context of RAG.

In this work, our goal is to explore the potential performance enhancement in relation extraction between entity pairs in a sentence through the use of a retrieval-augmented generation-based relation extraction (RAG4RE) approach¹, which leverages zero-shot settings. Specifically, we propose a pipeline for RAG-based relation extraction that utilizes open-source language models. To evaluate our RAG4RE, we leverage RE benchmark datasets, such as TACRED [19], TACREV [20], Re-TACRED [21] and SemEval [22] RE datasets. We utilize both encoder-decoder models, e.g., Flan T5 [23]—an instruction fine-tuned variant of the T5 model [24]—and decoder-only models like Llama2 [25] and Mistral [26], all of which are integrated into the approach outlined in Figure 2. Furthermore, we compare the performance of our RAG4RE approach with that of simple query (or vanilla) prompting to highlight how incorporating relevant contextual information into the prompt improves results and reduces false predictions.

In this work, our findings are:

1. The RAG-based RE (RAG4RE) approach has the potential to outperform both simple query (without relevant sentence), known as vanilla LLM prompting, and existing best-performing RE approaches from previous studies.

¹The source code is available at <https://github.com/sefeoglu/RAG4RE/>

2. While Decoder-only LLMs [27] still encounters hallucination issues on these datasets, our RAG4RE effectively mitigates this problem, especially when compared to the results obtained from the simple query.

In the following section, we first summarize recent works in RE and RAG in Section 2, and then provide a detailed description of the proposed RAG4RE in Section 3. We evaluate our RAG4RE on RE benchmark datasets, integrating different types of LLMs in Section 4. Next, we conduct ablation studies, which yield promising results for SemEval and provide inspiration for applying RAG4RE to domain-specific datasets in Section 5. Subsequently, Section 6 discusses RAG4RE’s results in comparison to those of previous approaches. Finally, we summarize the outcomes of our RAG4RE approach in Section 7.

2. Related Works

In this section, we summarize recent works into two categories: (i) Relation Extraction and (ii) Retrieval-Augmented Generation.

2.1. Relation Extraction

Relation Extraction (RE) is one of the main tasks of Information Extraction and plays a significant role among natural language processing tasks. RE aims to identify or classify the relations between entity pairs (head and tail entities).

RE can be carried out with various types of approaches: (i) supervised techniques including features-based and kernel-based methods, (ii) a special class of techniques which jointly extract entities and relations (semi-supervised), (iii) unsupervised, (iv) Open IE and (v) distant supervision-based techniques [4]. Supervised techniques require a large annotated dataset, and its annotation process is time-consuming and costly [4]. Distant supervision is amongst one of the popular methods dealing with the problem of obtaining annotated data. The distant supervision, based on existing knowledge bases, brings its own drawbacks, and it faces the issue of wrongly labeled sentences troubling the training due to the excessive amount of noise [5]. Another popular approach is weakly supervised RE [6]. However, the weakly supervised approach is more error-prone because of semantic drift in a set of patterns per iteration of its incremental learning approach like a snowball algorithm [6]. In rule-based RE approaches, finding relations is mostly restricted by predefined rules [4].

In terms of the best-performing RE approach, obtained by fine-tuning the language models, Cohen et al. [9] proposed a span-prediction-based approach for relation classification instead of single embedding to represent the relations between entities. This approach has improved the state-of-the-art scores on the well-known datasets. DeepStruct [7] proposed an innovative approach aimed at enhancing the structural understanding capabilities of language models. This work introduced a pre-trained model comprising 10 billion parameters, facilitating the seamless transfer of language models to structure prediction tasks. Specifically, regarding the RE task, the output format entails a structured representation of (head entity, relation, tail entity), while the input format comprises the input text along with a pair of head and tail entities. Zhou et al. [10] concentrated on addressing two critical issues that affect the performance of existing sentence-level RE models: (i) Entity Representation and (ii) noisy or ill-defined labels. Their approach extends the pretraining objective of masked language modeling to entities and incorporates a sophisticated entity-aware self-attention mechanism, enabling more accurate and robust RE. Li et al. [11] proposed a label graph to review candidate labels in the top-K prediction set and learn the connections between them. When predicting the correct label, they first compute that the top-K prediction set of a given sample contains useful information.

Zhang et al. [14] generated multiple-choice question prompts from test sentences where choices consist of verbalization of entities and possible relation types. These choices are selected from the training sentence based on entities in a test sentence. However, it could not outperform the previously introduced rule and ML -based approaches. In the context of their works, Zhang et al. [14] proved that enriching prompt context improves the prediction results on benchmark datasets such as TACRED and Re-TACRED. Melz [28] focuses on Auxiliary Rationale Memory for the RAG approach in the Relation Extraction task, and the proposed system learns from its successes without incurring high training costs. Chen et al. [8] proposes a Generative Context-Aware Prompt-tuning method which also tackles

the problem of prompt template engineering. This work proposed a prompt generator that is used to find context-aware prompt tokens by extracting and generating words regarding entity pairs and evaluated on four benchmark datasets: TACRED, TACREV, Re-TACRED and SemEval. Furthermore, Han et al. (2022) [12] employed prompt-tuning approaches as a mask-filling task, utilizing various encoders such as BART, RoBERTa, and the encoder component of T5-large on datasets like TACRED, ReTACRED, TACREV, and Wiki80. Their approach achieved F1 scores of 75.3% on TACRED and 84.0% on TACREV. However, the primary limitation of this work lies in the time efficiency of these autoregressive models.

In this work, we introduce a Retrieval-Augmented Generation-based Relation Extraction (RAG4RE) approach that operates in zero-shot settings to identify relations between entity pairs within a sentence. Previous approaches have been limited by their dependence on either labeled data during training [7–12] or by their use of prompt templates that lack sufficient contextual information [14]. In contrast, our RAG4RE method incorporates contextual information, reducing reliance on the potentially outdated internal knowledge of (vanilla) language models. A key advantage of our approach is that it eliminates the need for both a training process and a labeled dataset. We provide a detailed explanation of RAG in the next section.

2.2. Retrieval-Augmented Generation

Retrieval-Augmented Generation (RAG) for large language models can be classified into two categories: i) naive RAG and ii) advanced RAG. Naive RAG follows basic steps: retrieval, augmentation, and generation. In contrast, the advanced version incorporates post-processing steps, such as selecting essential information, condensing the context to be processed, and emphasizing critical parts of the retrieval context before delivering the retrieved information to the user [29]. The concept of RAG has been suggested as a way to minimize the undesired alterations in Language Models (LLMs) when conversational systems built on LLMs generate arbitrary responses to a query [13]. RAG is an example of open-book exams which are applied to the usage of LLMs. The retriever mechanism in RAG finds an example of the user query (prompt), and then the user query is regenerated along with the example by the data-augmentation module in RAG. Ovadia et al. [30] evaluates the knowledge injection capacities of both fine-tuning and the RAG approach and found that LLMs dealt with performance problems through unsupervised fine-tuning while RAG outperformed the fine-tuning approach in unsupervised learning.

3. Methodology

In this work, we have developed an RAG-based Relation Extraction (RAG4RE) approach to identify the relation between a pair of entities in a sentence. Our proposed RAG4RE, illustrated in Figure 2, consists of three modules: (i) Retrieval, (ii) Data Augmentation, and (iii) Generation. Our proposed RAG4RE approach is a variant of an advanced RAG [29], as its retrieval module includes “Result Refinement” which applies post-processing after responses from generation module. An example demonstrating the different responses returned to RAG4RE and a simple query is given in Table 11. The rest of the section explains the details of how each module of our proposed approach in Figure 2 works under specific subsections.

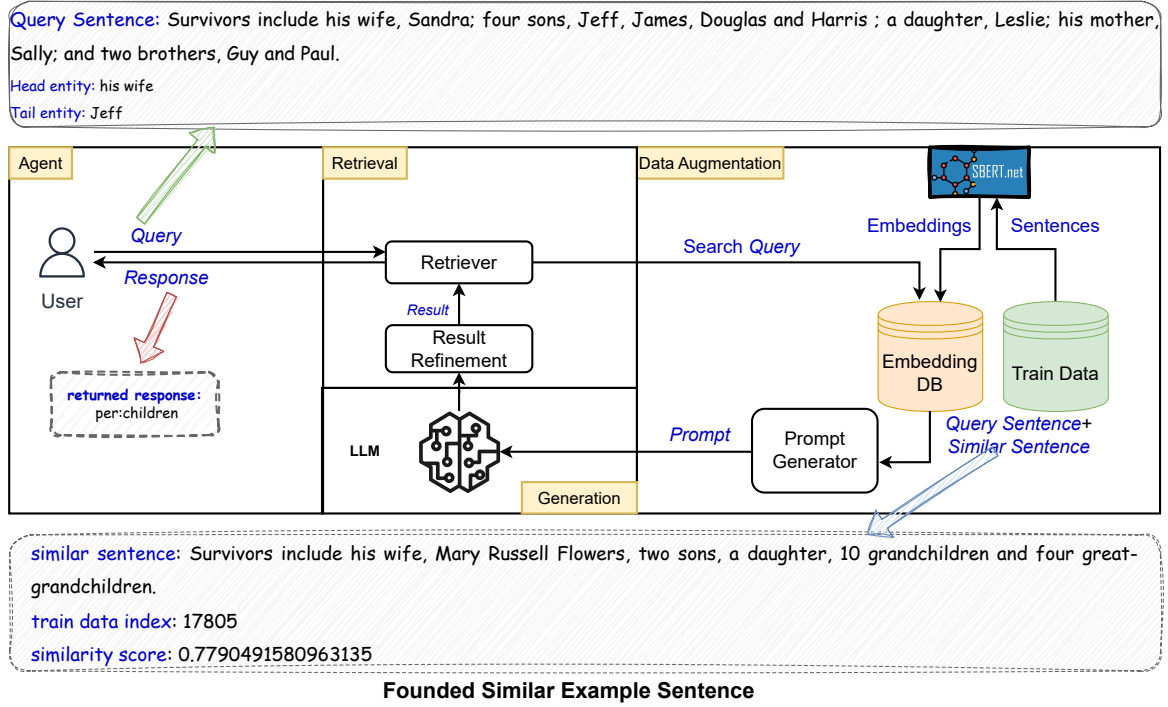


Fig. 2. RAG-based Relation Extraction (RAG4RE) pipeline, featuring a sample query sentence and the corresponding similar sentence retrieved.

3.1. Retrieval

A user submits a sentence (query) along with a pair of entities (head and tail entities) that might have a relation to the Retrieval module as demonstrated in Figure 2. Then, the Retriever sends this query to the Data Augmentation module, which extends the original query with a semantically similar sentence from training dataset, as an example given in Figures 2 and 4. “Result Refinement” in this module applies post-processing techniques, if necessary, to the results returned by the Generation module. The “Result Refinement” consists of a couple of response processing steps, such as refining prefixes (e.g., changing “per:member_of” to “org:member_of” as illustrated in Table 1 and Figures 8(a) and 8(b)), and converting “no relation” answers into “no_relation” as defined in the predefined relation types². Unfortunately, due to the nature of LLMs, which are based on next-token prediction, they might still generate undefined relation types, as analyzed in Section 4.3.

Table 1

Prefix refinement samples from Flan T5 XL, along with TACRED and its variants, based on predictions from the evaluation phase.

Raw Predicted Relations by LLMs	Refined Relation
org:religion	per:religion
per:member_of	org:member_of
org:employee_of	per:employee_of
org:schools_attended	per:schools_attended
per:members	org:members
org:parents	per:parents

²The raw results before processing: <https://github.com/sefeoglu/RAG4RE/tree/master/results/FlanT5/raw> .
 The processed results: https://github.com/sefeoglu/RAG4RE/tree/master/results/FlanT5/returned_responses.

3.2. Data Augmentation

The Data Augmentation module includes an Embedding Database (DB) containing embeddings of the training data, which are computed using the Sentence BERT (SBERT) model [31]. In our approach, we use the “all-MiniLM-L6-v2” version of SBERT³. Within this module, the embedding of the query sentence is also computed by SBERT. The system then calculates similarity scores between embeddings of each training sentence in the Embedding DB and the query sentence embeddings using the cosine similarity metric, as described in Equation (1). This formula measures the cosine similarity between two embedding vectors \mathbf{A} and \mathbf{B} , computed as the dot product of the vectors divided by the product of their magnitudes.

$$\text{Cosine Similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} \quad (1)$$

where \mathbf{A} and \mathbf{B} are the embedding vectors representing two different sentences in Equation (1). After computing the similarity scores between the query sentence embeddings and those in the Embedding DB, the system selects the sentence with the highest similarity (top one) and incorporates it into the prompt template, as shown in Figure 3. For example, the cosine similarity score between the query sentence, “*Survivors include his wife, Sandra; four sons, Jeff, James, Douglas, and Harris; a daughter, Leslie; his mother, Sally; and two brothers, Guy and Paul.*” and the top-ranked similar sentence, “*Survivors include his wife, Mary Russell Flowers, two sons, a daughter, 10 grandchildren, and four great-grandchildren.*” is approximately **0.77904**, as illustrated in Figure 2. Both the query sentence and the most similar sentence are then input into the prompt generator, which constructs the prompt based on this template. Essentially, the prompt generator reformulates the user query by including the relevant sentence from the Embedding DB. An example of the generated prompt is displayed in Figure 4. Additionally, no similarity threshold is set for retrieving the most similar sentence. Furthermore, the similarity computation based on embeddings does not guarantee that the most similar sentence will contain both the head and tail entities of the query sentence, as well as the same relation types between entities, since embeddings are computed for all tokens in a sentence. The generated prompt is then passed to the Generation module for further processing. Our prompt template (see Figures 3 and 4) incorporates possible relation types to leverage the conditional generation capabilities of LLMs.

Problem Definition: {Provide a clear and concise definition of the task (e.g., Relation Extraction), explaining its objective and significance.}
Question: {Pose a wh-question that pertains to the task, such as "What is the relationship between the head and tail entities in relation extraction?"}
Example Sentence: {A sentence that helps clarify the query, with embeddings similar to the query sentence. It will be retrieved from the Embedding Database based on cosine similarity, returning the most similar sentence.}
Query Sentence: {State the given sentence for which the task needs to be performed.}
Head Entity: {Identify the head entity within the sentence.}
Tail Entity: {Identify the tail entity within the sentence.}
Relation Types: {List all possible types of relationships that could be identified between the head and tail entities.}
Output Format: {Specify the desired output format, for example, the relation type between the head and tail entity, represented as "relation_type".}

Fig. 3. Illustration of the re-generated prompt template. The blue-colored Query Sentence, Head and Tail Entities, and Relation Types are provided by the user.

³The model is available at <https://huggingface.co/sentence-transformers/all-MiniLM-L12-v2>, access date: 09.02.2025

Problem Definition: Relation extraction is to detect the relationship between two entities in a sentence.
Question: What is the relation type between head and tail entities according to given relationships below in the following sentence?
Example Sentence:
 He is also an adviser to the Senatorial Committee of the US Congress and to the Democratic Party .
Query Sentence: Survivors include his wife, Sandra; four sons, Jeff, James, Douglas and Harris ; a daughter, Leslie; his mother, Sally; and two brothers, Guy and Paul.
Head entity: his wife
Tail entity: Jeff
Relation types: org:alternate_names, org:stateorprovince_of_headquarters, per:cause_of_death, per:age, per:cities_of_residence, per:siblings, per:parents, org:member_of, per:stateorprovince_of_birth, per:date_of_death, per:schools_attended, per:city_of_death, no_relation, per:employee_of, org:subsidiaries, per:alternate_names, per:city_of_birth, per:title, per:stateorprovince_of_death, org:city_of_headquarters, org:website, per:country_of_birth, org:founded_by, org:founded, org:top_members/employees, per:other_family, org:members, org:number_of_employees/members, per:spouse, per:date_of_birth, per:countries_of_residence, per:stateorprovinces_of_residence, per:children, per:religion, per:charges, org:parents, org:political/religious_affiliation, per:origin, org:dissolved, org:country_of_headquarters, org:shareholders, per:country_of_death.
output format: relation_type

Fig. 4. An example of a re-generated prompt from a sample in the TACRED dataset.

3.3. Generation

The LLM generates a response for the prompts using zero-shot settings in the Generation module. We integrate LLMs with different architectures, including encoder-decoder and decoder-only models [27], in our experiments so that we can evaluate the performance of our proposed RAG4RE approach with different LLMs and compare them within the RAG4RE framework. Subsequently, the response is sent to the “Result Refinement” in the Retrieval module. Result refinement might be necessary if the relation type includes a prefix, as the response might omit or incorrectly predict the prefix. For example, the response might return *member_of* or *per:member_of* instead of *org:member_of* as given in Table 1⁴ (See Figures 8(a) and 8(b) for statistics about prefix refinement when Flan T5 has been integrated into our RAG4RE pipeline.). The Generation module concludes when the results are sent to the Retrieval module. Afterwards, the “Retriever” sends the results to the user. An example of the responses returned by the retriever is shown in Figure 2.

4. Evaluation

In this section, we examine the performance of our RAG4RE work. We first introduce the experimental settings in Section 4.1. Then, we present the results of our experiments in Section 4.2. Finally, false predictions are analyzed in Section 4.3.

4.1. Experimental Setup

In our work, we assess the effectiveness of our RAG4RE approach using well-established RE benchmarks, including the TACRED [19], TACREV [20], Re-TACRED [21], and SemEval [22] RE datasets. These benchmarks consist of head and tail entities in the given sentences, along with the ground truth relation types between these entities. These widely recognized datasets serve as invaluable resources for evaluating the efficiency and performance of our approach. Further insights into the datasets can be found below and in Section 4.1.1. Additionally, we compare the performance of our RAG4RE approach, which incorporates a relevant (similar) sentence alongside a query sentence in its prompt template (see the prompt template in Figure 3), to that of a simple query prompt—referred to as the vanilla prompt—which excludes any relevant sentence related to the query sentence, as explored in previous works [14]. This comprehensive evaluation helps assess our approach’s performance across different LLMs.

⁴The refinements in Table 1 can be checked in the analysis folder: <https://github.com/sefeoglu/RAG4RE/tree/master/results/FlanT5/analysis>

4.1.1. Datasets

We utilize four benchmark datasets, as detailed below and in Table 2. Figures 6 and 7 provide statistics for the SemEval RE dataset, while Table 3 offers details about the ‘no_relation’ type in TACRED and its variants.

- **TACRED**, namely the TAC RE Dataset, is a supervised RE dataset obtained via crowdsourcing and targeted towards TAC KBP relations. There is no directionality in predefined relations, which can also be extracted from a given sentence tokens. We directly used this licensed dataset from Linguistic Data Consortium (LDC)⁵.
- **TACREV** is a revisited version of TACRED that reduces noise in sentences defined with “no_relation”. In our work, this dataset is generated from original TACRED by running source codes given at [20]⁶.
- **Re-TACRED** is a re-annotated version of the TACRED dataset that can be used to perform reliable evaluations of RE models. To generate this Re-TACRED from original TACRED, we leverage source codes⁷ given at [21].
- **SemEval**: focuses on multi-way classification of semantic relationships between entity pairs. The predefined relations (target relation labels) in this benchmark dataset have directions and cannot be extracted from tokens of (test, or train) sentences in the dataset. Figures 6 and 7 demonstrate details about this dataset. The dataset is obtained from HuggingFace⁸.

Table 2
Overview of benchmark datasets. ‘-’ indicates the absence of a validation split.

Split	TACRED	TACREV	Re-TACRED	SemEval
Training	68124	68124	58465	8000
Test	15509	15509	13418	2717
Validation	22631	22631	19584	-
# of Relations	42	42	40	19

Table 3
Statistics of relation type ‘no_relation’ across TACRED and its variants.

Dataset	Total Test	# of no_relation	Percent of no_relation
TACRED	15,509	12,184	78.56%
TACREV	15,509	12,386	79.86%
ReTACRED	13,418	7,770	57.91%

4.1.2. Pre-Trained Language Models

We evaluate our RAG4RE approach on the aforementioned benchmark datasets by integrating various LLMs, including Flan T5 (XL⁹ and XXL¹⁰), Llama-2-7b-chat-hf¹¹, and Mistral-7B-Instruct-v0.2¹². We use the instruction fine-tuned versions of Llama2 and Mistral, as Flan T5 [23] is itself an instruction fine-tuned version of the T5 model [24].

⁵The TAC Relation Extraction dataset catalog is accessible at <https://catalog.ldc.upenn.edu/LDC2018T24>, access date: 24.01.2025

⁶The TACREV source code repository is available at <https://github.com/DFKI-NLP/tacrev>, access date: 24.01.2025.

⁷The Re-TACRED source code are at <https://github.com/gstoica27/Re-TACRED>, access date: 24.01.2025.

⁸The SemEval Dataset card: https://huggingface.co/datasets/sem_eval_2010_task_8, access date: 24.01.2025.

⁹Flan T5-XL:<https://huggingface.co/google/flan-t5-xl>, access on 24.01.2025

¹⁰Flan T5 XXL:<https://huggingface.co/google/flan-t5-xxl>, access on 24.01.2025

¹¹Llama-2-7b-chat-hf:<https://huggingface.co/meta-llama/Llama-2-7b-chat-hf>, access on 24.01.2025

¹²Mistral-7B-Instruct-v0.2: <https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2>, access on 24.01.2025

4.1.3. Evaluation Metrics

We compare our RAG4RE approach with simple query, vanilla prompting, in terms of micro F1, Recall, and Precision scores, as given in Equations (2) to (4). In these equations, True Positive, False Positive and False Negative are denoted as TP, FP and FN, respectively, where n in Equations (2) to (4) points out total number of classes or categories, and i is an index representing a class or category in a multi-class classification problem, due to the imbalance in these benchmark datasets (See Table 3). To compute these metrics, we leverage metrics library of sklearn¹³.

$$\text{Micro Precision} = \frac{\sum_{i=1}^n \text{TP}_i}{\sum_{i=1}^n (\text{TP}_i + \text{FP}_i)} \quad (2)$$

$$\text{Micro Recall} = \frac{\sum_{i=1}^n \text{TP}_i}{\sum_{i=1}^n (\text{TP}_i + \text{FN}_i)} \quad (3)$$

$$\text{Micro F1} = \frac{2 \cdot \text{Micro Precision} \cdot \text{Micro Recall}}{\text{Micro Precision} + \text{Micro Recall}} \quad (4)$$

4.1.4. Hardware Details

In regard to hardware specifications, these language models have undergone evaluation utilizing a setup comprising 4 GPUs, with each GPU boasting a memory capacity of 12 GB in the NVIDIA system. Device details are NVIDIA GeForce GTX 1080 Ti (4GPUs X 12GB). Furthermore, the memory configuration reaches 300 GB.

4.2. Results

Our experiments are conducted using the four benchmark datasets mentioned in Section 4.1.1. Firstly, we assess the performance of our proposed RAG4RE and then compare it to that of a simple query (sentence), vanilla LLM prompting, in terms of micro F1 score. As mentioned earlier, our evaluation criteria take into account the micro F1 score, Recall, and Precision metrics due to the imbalanced labelling of the datasets (See Table 3). Furthermore, we explore how our approach enhances the performance of LLM responses. This is accomplished by incorporating the example sentence which is the most similar to the query sentence—determined using the cosine similarity metric at Equation (1) with SBERT embeddings—into the prompt template alongside the query sentence in our proposed RAG4RE approach (see Data Augmentation in Figure 2). We compare the results of a simple query without any relevant sentence to our RAG4RE results at Table 4.

We utilize various LLMs, including Flan T5 XL and XXL, Mistral-7B-Instruct-v0.2, and Llama-2-7b-chat-hf, to conduct our experiments and evaluate the performance of our proposed RAG4RE framework. The results demonstrate that RAG4RE achieves remarkable performance compared to a simple query approach, as shown in Table 4 and Figure 9. Notably, RAG4RE consistently outperforms the simple query approach across the TACRED, TACREV, and Re-TACRED datasets, even when the underlying language model is varied. The highest F1 scores achieved, as detailed in Table 4, are 86.6%, 88.3%, and 73.3% for TACRED, TACREV, and Re-TACRED, respectively. These remarkable results are primarily accomplished by integrating the Flan T5 XL model into the Generation module. Nonetheless, RAG4RE does not achieve comparable performance on the SemEval dataset. This might be primarily due to either the predefined relations (target relation labels) in this dataset, which cannot be directly extracted from the sentence tokens, or the lack of knowledge about this dataset in the vanilla LLMs used in RAG4RE. Furthermore, the SemEval dataset includes manually annotated sentences for specific, defined semantic relation types [22].

The remarkable improvement observed in RAG4RE’s results can be primarily attributed to the incorporation of relevant (or similar) example sentences, extracted from the training data of benchmark datasets, into the user query

¹³Sklearn evaluation library: https://scikit-learn.org/stable/modules/model_evaluation.html

Table 4

Experimental results on four benchmark datasets using different LLMs. The best results are highlighted in orange. For comparison, prior state-of-the-art (SoTA) and LLM-based results are included, with top results marked in blue. ‘-’ indicates missing results for that metric on the corresponding dataset.

Our Results													
LLM	Method	TACRED			TACREV			Re-TACRED			SemEval		
		P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)
Flan T5 XL	simple query	91.0	79.2	84.7	97.2	49.0	65.1	69.5	73.0	71.2	20.41	12.32	15.37
	RAG4RE	84.5	88.8	86.6	84.5	92.4	88.3	63.0	87.7	73.3	17.16	11.93	14.07
Flan T5 XXL	simple query	94.9	44.7	60.9	98.2	27.1	42.4	70.2	39.3	50.4	13.64	13.4	13.52
	RAG4RE	93.3	61.0	73.8	91.7	82.3	86.7	78.0	53.6	63.5	17.32	15.39	16.29
Llama-2-7b-chat-hf	simple query	84.97	1.21	2.38	74.64	0.44	0.87	80.2	0.94	1.86	5.89	5.08	5.45
	RAG4RE	81.23	55.01	65.59	84.89	54.57	66.43	55.93	3.46	6.52	4.36	4.2	4.28
Mistral-7B-Instruct-v0.2	simple query	94.67	11.96	21.23	92.34	5.15	9.75	64.64	5.48	10.11	25.5	24.37	24.92
	RAG4RE	87.81	30.1	44.83	93.23	22.59	36.36	60.19	30.08	40.11	24.1	22.75	23.41
Results from SoTA and LLM-based Approaches													
SoTA	DeepStruct [7]	-	-	76.8	-	-	-	-	-	-	-	-	-
	EXOBRAIN [10]	-	-	74.6	-	-	83.2	-	-	91.1	-	-	-
	KLG [11]	-	-	75.6	-	-	84.1	-	-	-	-	-	90.5
	SP [9]	-	-	74.8	-	-	-	-	-	-	-	-	91.9
LLM-based	GAP [8]	-	-	72.7	-	-	82.7	-	-	91.4	-	-	90.3
	LLMQA4RE [14]	-	-	52.2	-	-	53.4	-	-	66.5	-	-	43.5
	RationaleCL [15]	-	-	80.8	-	-	-	-	-	-	-	-	-

sentence. As highlighted in [13], RAG operates akin to an open-book exam, where adding a relevant (or similar) sentence to the query sentence facilitates the LLM’s understanding of the query sentence in the Generator module of our approach, as shown in Figure 2. The example sentence and query sentence might have similar or same entities, as illustrated in Figure 2, which helps the LLM make accurate inferences and reduce hallucinations. This interpretation is further supported by the results of the simple query and RAG4RE approaches, as outlined in Table 4.

Consequently, our RAG4RE has improved F1 scores on benchmark datasets, e.g., TACRED, TACREV and Re-TACRED, when compared its results to those of a simple query as demonstrated in Table 4. This performance improvement between simple query and RAG4RE can be explained by spread of activation theory [32]. In terms of embeddings, similar sentences might contain the same or semantically similar (or closely related) entities as those in the query sentence. In this context, similar sentences serve as facilitators for comprehending the entities in the query sentence. This role can be explained by the spread of activation theory, which describes how a computational model activates one word (token) and spreads its influence to related words or concepts in the model [32]; for instance, “sky” reminds one of “blue” or “cloud” as explained by the spread of activation theory. In the next section, we closely analyze the prediction errors on TACRED, its variants, and SemEval for further insights about how RAG4RE works.

4.3. Error Analysis

In this section, we analyze how RAG4RE improves the results of simple query (sentence) across three benchmarks, whereas it could not demonstrate this improvement on SemEval dataset. We mainly discuss false predictions and undefined relation types predicted by LLMs.

Decoder only LLMs, e.g., Mistral and Llama2, are prone to producing hallucinatory results when a simple prompt is sent to those models [27]. In the responses of the experiments conducted with Mistral-7B-Instruct-v0.2 and Llama-2-7b-chat-hf, we observed such relation types that have not been defined in the relation types of the prompt template (See the example prompt in Figure 3). Therefore, we analyze these undefined relation types generated by LLMs in Figure 5 across TACRED and its variants. According to Figure 5, both decoder-only models used in our experiments generate more undefined relation types than encoder-decoder models, Flan T5 XL and XXL.

We also analyze the False Negative (FN) and False Positive (FP) relation predictions of three language models in Table 5. In terms of FN predictions, our RAG4RE has decreased its FN predictions on the Flan T5 XL model in Table 5. Likewise, our RAG4RE has reduced the number of FN predictions on Mistral-7B-Instruct-v0.2 and Llama-2-7b-chat-hf. Regarding dataset insights, the reason Flan T5 XL performs better on the TACREV and TACRED datasets but underperforms on ReTACRED is mainly due to the number of ‘no_relation’ labels in these datasets (see Table 6). ReTACRED is a reannotated version of TACRED created using a source code, while TACRED is noisier and less reliable. The relation type ‘no_relation’ in the ReTACRED dataset makes up 57.91% of the overall relation types in its test dataset, and Flan T5 XL generates 80.69% ‘no_relation’ predictions among all test relations. This means it generates ‘no_relation’ predictions as frequently as the proportion of ‘no_relation’ in the TACRED test dataset. This might be related to the data used in the training phase of Flan T5 models, as TACRED is a crawling dataset and the base model, T5, of Flan T5 was trained on a crawling dataset¹⁴. Additionally, the decrease in FNs is greater than the increase in FPs in most cases in Table 5.

With regards to the SemEval dataset, the increase in FPs is higher than the decrease in FNs when RAG4RE is used with Flan T5 XL, meaning the results are not improved. Similarly, the number of FNs has increased with decoder-only models when RAG4RE is used instead of a simple query, so the results are not improved. Additionally, the SemEval dataset is partially manually annotated and not a web crawling dataset; therefore, the vanilla LLMs might not have prior knowledge of this dataset.

Overall, when comparing the number of false predictions, both Llama-2-7b-chat-hf and Mistral-7B-Instruct-v0.2 produce higher numbers compared to Flan T5 XL. It is clear that RAG4RE mitigates hallucination problem of the simple query by reducing the number of false prediction on three language model types according to Table 5 on TACRED and its variants.

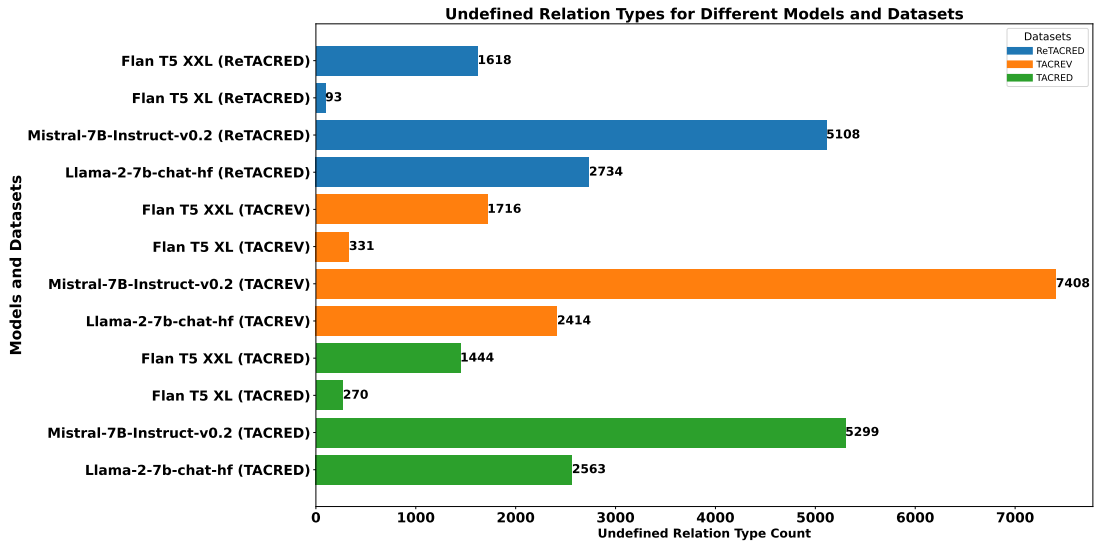


Fig. 5. Gives number of undefined relation predictions across TACRED and its variants on different LLM along with RAG4RE. These relation types are not defined in the relation types in datasets (See Table 2).

¹⁴C4 dataset: <https://paperswithcode.com/dataset/c4>

Table 5

Comparison of False Positives (FP) and False Negatives (FN) between the Simple Query and RAG4RE approaches across different LLMs. The table highlights changes in FP (indicated in blue) and FN (indicated in orange). Increases and decreases in FP and FN are denoted by + and -, respectively.

LLM	Dataset	FP		FN		Increase (+) /Decrease (-)	
		Simple Query	RAG4RE	Simple Query	RAG4RE	FP Changes	FN Changes
<i>Flan T5 XL</i>	TACRED	956	1990	3726	1987	+1034	-1739
	TACREV	316	2337	8180	1390	+2021	-6790
	Re-TACRED	2481	4013	3676	1735	+1532	-1941
	SemEval	1306	1564	1076	829	+258	-247
<i>Llama-2-7b-chat-hf</i>	TACRED	26	1549	15336	7258	+1523	-8078
	TACREV	18	1184	15438	7676	+1166	-7762
	Re-TACRED	18	212	13327	12937	+194	-390
	SemEval	0	0	2579	2603	No Change	+24
<i>Mistral-7B-Instruct-v0.2</i>	TACRED	82	509	13970	11333	+427	-2637
	TACREV	52	200	14830	12557	+148	-2273
	Re-TACRED	233	1546	12759	9535	+1313	-3224
	SemEval	0	0	2055	2099	No Change	+44

Table 6

Statistics of no_relations prediction across TACRED and its variant along with Flan T5 XL for RAG4RE.

Dataset	Total Test	# no_relation predicted	% no_relation predicted	% no_relation in test dataset	F1 of no_relation
TACRED	15,509	12,811	82.60%	78.56%	87%
TACREV	15,509	13,548	87.36%	79.86%	88%
ReTACRED	13,418	10,827	80.69%	57.91%	73%

5. Ablation Study

In this section, we conduct ablation studies: (i) Prompt Engineering approaches, (ii) Recent Advancements in Llama models and (ii) Post-Training Approaches on SemEval. We evaluated these experimental approaches on original TACRED and SemEval datasets¹⁵.

5.1. Prompt Engineering Approaches

This section evaluates two approaches: (i) one-shot prompting and (ii) another prompt template for RAG4RE. We first present the results of one-shot prompting in the following section, and then discuss the impact of varying prompt templates on the results.

5.1.1. One-Shot Prompting

In this section, we conducted an experiment using one-shot prompting as a prompt engineering approach with the original TACRED dataset and SemEval, alongside the previously used Flan T5 models (XL and XXL). This experiment departs from zero-shot settings and includes an example. We identified similar sentences and incorporated them into our prompt templates. For this section, we include head and tail entities, as well as their relation type, in the sample prompt template, as shown in Figure 3 and Figure 10. The results in Table 7 show that the one-shot prompting approach cannot outperform RAG4RE when using the Flan T5 (XL and XXL) models on the TACRED

¹⁵The experiments in this section conducted on Google Pro+ A100 GPU.

and SemEval datasets, as indicated in Table 4. This one-shot approach also fails to achieve the performance of a simple query on TACRED (see Table 4). However, one-shot prompting with Flan T5 XL and XXL on SemEval improves micro F1 by 16.44% and 38.60%, respectively.

Table 7
One-shot experiment on TACRED and SemEval

LLM	TACRED			SemEval		
	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)
<i>Flan T5 XL</i>	81.36	70.90	75.77	21.75	13.21	16.44
<i>Flan T5 XXL</i>	74.36	49.99	59.79	38.830	38.37	38.60

5.1.2. Another Prompt Template for RAG4RE

We also evaluate our RAG4RE system using a different prompt template that excludes the problem definition, as shown in Figure 3, on the TACRED and SemEval datasets. However, we do not evaluate this prompt template, shown in a sample in Figure 11, on TACREV or Re-TACRED, since they are variants of the TACRED dataset. Excluding the problem definition from the prompt template resulted in a decrease in the F1 score from 86.6% to 82.89% for RAG4RE with Flan T5 XL, and a similar drop of 4.71% was observed for simple query results on TACRED. In contrast, on the SemEval dataset, this prompt template resulted in an improvement, increasing the micro F1 score by 18.91% for simple queries and by 17.42% for RAG4RE with Flan T5 XL. These results indicate that the effectiveness of the prompt template is highly dependent on the dataset. While the choice of prompt template can significantly enhance or diminish RAG4RE performance, RAG4RE consistently outperformed simple queries on TACRED across both prompt templates, along with Flan T5. Similar to the previously used prompt template, no performance improvement was observed on SemEval with this prompt template when using the RAG4RE system.

Table 8
Another prompt template which does not have any problem definition.

LLM	Method	TACRED			SemEval		
		P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)
<i>Flan T5 XL</i>	simple query	93.66	69.81	79.99	32.86	35.82	34.28
	RAG4RE	88.7	77.8	82.89	38.91	26.44	31.49
<i>Llama-2-7b-chat-hf</i>	simple query	32.13	19.52	24.29	26.51	26.59	26.55
	RAG4RE	34.19	13.08	18.92	17.20	16.83	17.01
<i>Mistral-7B-Instruct-v0.2</i>	simple query	20.66	11.19	14.52	100.0	0.04	0.08
	RAG4RE	23.45	11.37	15.31	57.89	0.44	0.88

5.2. Comparing Llama Variants

We evaluate a different instruction-tuned version of the Llama model: Llama-3.1-8B-Instruct¹⁶. This model has more parameters than the one used to produce the results shown in Table 4. Our goal is to examine whether the proposed approach behaves differently when using a model with a larger parameter count. As shown in Table 9, our RAG4RE approach outperforms the simple prompting baseline on the TACRED, TACREV, and Re-TACRED datasets when using Llama-3.1-8B-Instruct within our framework (illustrated in Figure 2), similar to the results obtained with Llama-2-7b-chat-hf. For the SemEval dataset, the simple query performs slightly better, though the results between the simple query and RAG4RE remain close, as seen in Table 9. Beyond internal comparisons,

¹⁶Llama-3.1-8B-Instruct is available at <https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct>, access on 9.06.2025.

Llama-3.1-8B-Instruct with the simple query achieves higher performance than Llama-2-7b-chat-hf on four benchmark datasets (see Table 4 and Table 9). However, when using the RAG4RE approach, Llama-3.1-8B-Instruct does not surpass Llama-2-7b-chat-hf on TACRED and TACREV.

Table 9
The experimental results on four benchmark datasets along with Llama3.1-8B

LLM	Method	TACRED			TACREV			Re-TACRED			SemEval		
		P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)
Llama-3.1-8B-Instruct	simple query	37.12	24.06	29.19	30.89	18.79	23.36	27.10	17.61	21.35	27.95	27.94	27.94
	RAG4RE	49.90	46.75	48.27	62.33	54.52	58.16	36.24	30.83	33.32	27.72	27.71	27.72

5.3. Post-Training Approaches on SemEval Dataset

We propose that fine-tuning LLMs on a small subset of the SemEval dataset could enhance model performance and better adapt the LLMs to the domain of the dataset. Therefore, we fine-tune the Flan T5 Base (250M) model on two distinct versions of SemEval dataset: one based on the simple queries introduced in previous sections (see Table 11 for an example), and the original SemEval sentence dataset, where the input is a sentence and the output is a relation type. The subset is selected from the remaining data after identifying the sentences most similar to the test sentences in the training dataset. Fine-tuning is then performed on 5,283 samples (train and validation) from the SemEval training dataset using 5-fold cross-validation¹⁷. The hyperparameters are set to 5 epochs, a batch size of 16, and a learning rate of 0.001 along Low-Rank Adaptation (LoRA) [33]. $LoRA_\alpha$ is set to 32, the rank parameter (r) is 4, the task type is Seq2SeqLM, and $LoRA_{dropout}$ is 0.01 for the Flan T5 Base model.

We first fine-tune the model on the original dataset¹⁸ (See Table 12 for a sample from SemEval dataset.) and evaluate its performance using simple queries and RAG4RE, as shown in Table 10. Afterwards, we also fine-tune the Flan T5 Base on the prompt datasets¹⁹ and conduct the same evaluation approach along with simple query prompt and RAG4RE. According to the results in Table 10, our RAG4RE, utilizing these fine-tuned Flan T5 Base models, outperforms the simple query approach. As a result, these findings might provide inspiration for evaluating domain-specific datasets with RAG4RE.

Table 10
Mean metrics of the fine-tuned Flan T5 Base models across 5 runs with 5-fold cross-validation.

Dataset	Method	P (%)	R (%)	F1 (%)
Prompt Dataset from SemEval	fine-tune +Simple Query	78.01	70.33	73.89
	fine-tune + RAG4RE	75.65	75.65	75.65
Original SemEval sentences	fine-tune +Simple Query	36.15	36.15	36.15
	fine-tune +RAG4RE	38.37	38.37	38.37

6. Discussion

In our experiments, we compared two methods: (i) RAG-based Relation Extraction (RAG4RE) and (ii) a simple query, vanilla LLM prompting, which lacks inclusion of the relevant example sentence described in Figure 3. Our findings indicate a notable enhancement in F1 scores when employing the RAG4RE approach over the simple query

¹⁷The prompt dataset splits are available: https://huggingface.co/datasets/Sefika/semeval_prompt_dataset_splits

¹⁸The models are available at <https://huggingface.co/collections/Sefika/semeval-base-ft-67a7cd43d6ea7ddc90d33ace>

¹⁹The models are available: <https://huggingface.co/collections/Sefika/semeval-instruction-finetuned-models-67a4e41c20664ef74531a01b>

method on the TACRED dataset and its variations, as outlined in Table 4. This improvement stems from the integration of a relevant sentence into the prompt template, as illustrated in Figure 3. This inclusion of relevant example sentence facilitates the predictions made by the LLM in the Generation module of our proposed architecture, as depicted in Figure 2.

The incorporation of this relevant sentence serves to mitigate hallucinations in the LLM’s responses, subsequently reducing the occurrence of false predictions, as demonstrated in Table 5. Additionally, while RAG4RE enhances the generation capabilities of LLM models, they might still produce hallucinated relation types. Figure 5 gives the count of undefined relations in predictions across datasets and LLMs. Mistral and Llama models generate a significant number of undefined relations—relations that do not exist in the predefined set of relation types in the given prompt. In contrast, the Flan T5 XL model produces the lowest undefined relations. Our assessment of the RAG4RE approach’s effectiveness is based on the integration of Flan T5 XL into the LLM within Figure 2, given that our approach, combined with Flan T5 XL, yields the highest F1 scores across benchmarks except for the SemEval (See Table 4). Although the prompt tuning approach using a mask filling template is applied with T5 Large (fine-tuning its encoder) and evaluated on TACRED [12] (achieving an F1 score of 75.3%), its performance could not outperform that of Flan T5 XL with RAG4RE. This discrepancy in performance might be related to the size of the language models and the fact that only the encoder part of the T5 model is fine-tuned in [12].

The large margin between Re-TACRED and TACREV, or Re-TACRED and TACRED, is due to the percentage of sentences in their test datasets where the relation type between the given entities is ‘no_relation’ in RAG4RE approach. While TACRED and TACREV have high percentages of 78.56% and 79.86%, respectively, with Flan T5 XL predicting a high number of ‘no_relation’ sentences, Re-TACRED has only 57.91% of its test sentences labeled as ‘no_relation’, with Flan T5 XL predicting more than existing percent of no_relation (80.69%) (See Table 6). Furthermore, Re-TACRED evaluates 40 relation types, whereas TACREV and TACRED evaluate 42 relation types. Another important reason why Flan T5 XL does not perform better on ReTACRED might be related to that Flan T5 whose base model, T5, was trained on the C4 (Colossal Clean Crawled Corpus) dataset²⁰ which was constructed from free public data resource as TACRED dataset [19] was constructed. Additionally, ReTACRED is reannotated with a codebase from TACRED and is not available on the web. Nevertheless, RAG4RE improves performance of the simple query approach on TACRED and its variant even if Flan T5 models might know dataset insight for TACRED. Likewise, fine-tuning Flan T5 with a small amount of the SemEval dataset could improve the performance of both RAG4RE and simple queries, as it would adapt the model to this specific dataset as shown in the ablation study in Section 5.3.

We present an analysis of the performance of our RAG4RE approach, comparing its F1 score with both LLM-based methods and state-of-the-art RE techniques reported in current literature. In terms of LLM-based RE approaches, our RAG4RE consistently outperforms other methods utilizing LLMs, as illustrated in Table 4, across all benchmark datasets except for SemEval. The reason for the superior performance of our RAG4RE, as presented in Table 4, is largely attributed to the absence of relevant sentence addition in the prompt templates of both LLMQA4RE [14] whose prompt template is based on multiple-choice question and RationaleCL [15] (with F1 of 80.8% on TACRED) based on conversational prompting based on rational strategy. Notably, neither competing method incorporated relevant sentences into their prompt templates. These results are further supported by Min et al. (2022) [34], who compared the performance of multiple-choice tasks with classification tasks, demonstrating that language models perform better in classification tasks than in multiple-choice templates. Additionally, LM-MQA4RE does not include the prefix of the relation in its TACRED and its variants’ prompt templates, so there is no need to address incorrect prefix predictions. In contrast, we incorporate the prefix in our RAG4RE architecture. The changes made to the prefix during the integration of Flan T5 (XL and XXL) can be observed in Figure 8. In addition to the performance comparison with LLM-based approaches, we also compare our results with the best-performing methods in the literature, as shown in Table 4. Our RAG4RE outperformed all state-of-the-art approaches, including recently proposed models, fine-tuned language models, and advanced techniques, on both TACREV and TACRED, achieving F1 scores of 86.8% and 88.3%, respectively. However, it did not achieve the same performance on Re-TACRED, primarily due to the high number of ‘no_relation’ predictions, as detailed in Table 6. Similarly, due to

²⁰See dataset and this information at <https://paperswithcode.com/dataset/c4>, access on 03.02.2025.

the unique features of the SemEval dataset—such as directed relations and relations that cannot be predicted from sentence tokens, or was not used in the training of the vanilla LLMs used at Table 4—our RAG4RE did not yield promising results on this dataset. For example, the LLM-based approach, GAP [8], fine-tunes the RoBERTa large model (355M parameters) using a prompting strategy and achieves an F1 score of 90.3%. Likewise, our Flan T5 Base (250M) model fine-tuned on a subset of the SemEval train dataset improves the results of RAG4RE in Section 5.3.

With regard to ethical considerations, our proposed approach is evaluated on local hardware using open-source models. Therefore, no data is shared outside the local hardware. Furthermore, this study does not involve any human subject data, and thus does not have ethical concerns related to human data use. Nonetheless, the approach is designed to preserve privacy and can be applied to the evaluation of sensitive domain data, such as in the healthcare sector.

Overall, our RAG4RE demonstrates strong performance on the TACRED and TACREV datasets. However, its performance on SemEval does not achieve similar improvements, likely due to challenges posed by the predefined relation types (target relation labels) in this benchmark dataset or the limitations of vanilla LLMs’ prior knowledge (see results in Section 5.3). For example, directly extracting the “Cause-Effect (e2,e1)” relation type from the provided sentence tokens between entity 1 (e1) and entity 2 (e2) remains challenging for zero-shot LLM prompting, as this relation type often requires logical inference for accurate identification.

7. Conclusion and Future Work

In this work, we introduce a novel approach to Relation Extraction (RE) called Retrieval-Augmented Generation-based Relation Extraction (RAG4RE) which leverages zero-shot prompting settings. Our aim is to identify the relation types between head and tail entities in a sentence, utilizing an RAG-based LLM prompting approach.

We also claim that RAG4RE has outperformed the performance of the simple query (vanilla LLM prompting). To prove our claim, we conducted experiments using four different RE benchmark datasets: TACRED, TACREV, ReTACRED, and SemEval, in conjunction with three distinct LLMs: Mistral-7B-Instruct-v0.2, Flan T5 (XL and XXL), and Llama-2-7b-chat-hf. Our RAG4RE yielded remarkable results compared to those of the simple query. Our RAG4RE exhibited notably results on the benchmarks compared to previous works. Unfortunately, our proposed methods, including vanilla LLMs, did not perform well on the SemEval dataset. This can be attributed either to the absence of logical inference in LLMs or to the lack of prior knowledge in vanilla LLMs regarding this dataset, as predefined or target relation types cannot be directly derived from the sentence tokens in the SemEval dataset. The ablation study conducted with SemEval in Section 5.3 yields promising results when the post-trained model is integrated into RAG4RE. These findings may also encourage the application of RAG4RE to domain-specific datasets. In addition to the results presented in Section 5.3, Llama-3.1-8B-Instruct outperforms Mistral-7B-Instruct-v0.2, Flan T5 (XL and XXL), and Llama-2-7b-chat-hf on the SemEval dataset. These findings suggest that larger models may serve as more effective domain experts compared to their smaller counterparts.

In future work, we aim to extend our approach to real-world dynamic learning scenarios, inspired by the ablation study on SemEval, and evaluate it on real-world datasets. Additionally, we intend to integrate fine-tuned LLMs on training datasets into our RAG4RE system to address the performance issues encountered when datasets require logical inference to identify relation types between entities in a sentence, and target relation types cannot be extracted from the sentence tokens as in SemEval.

References

- [1] R. Grishman, Information Extraction, *IEEE Expert* **30**(5) (2015), 8–15.
- [2] S. Efeoglu, A continual relation extraction approach for knowledge graph completeness (short paper), 26th International Conference on Theory and Practice of Digital Libraries (TPDL) CEUR Workshop, Padua, Italy, 2022. https://ceur-ws.org/Vol-3246/15_paper110.pdf.
- [3] S. Pawar, G.K. Palshikar and P. Bhattacharyya, Relation Extraction : A Survey, 2017. <https://arxiv.org/abs/1712.05191>.
- [4] S. Pawar, G.K. Palshikar and P. Bhattacharyya, Relation Extraction : A Survey, arXiv, 2017.
- [5] M. Aydar, O. Bozal and F. Ozbay, Neural relation extraction: a survey, arXiv, 2020.

- [6] E. Agichtein and L. Gravano, Snowball: Extracting Relations from Large Plain-Text Collections, in: *Proceedings of the Fifth ACM Conference on Digital Libraries*, DL '00, Association for Computing Machinery, New York, NY, USA, 2000, pp. 85–94. ISBN 158113231X.
- [7] C. Wang, X. Liu, Z. Chen, H. Hong, J. Tang and D. Song, DeepStruct: Pretraining of Language Models for Structure Prediction, in: *Findings of the Association for Computational Linguistics: ACL 2022*, S. Muresan, P. Nakov and A. Villavicencio, eds, Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 803–823. doi:10.18653/v1/2022.findings-acl.67. <https://aclanthology.org/2022.findings-acl.67>.
- [8] Z. Chen, Z. Li, Y. Zeng, C. Zhang and H. Ma, GAP: A novel Generative context-Aware Prompt-tuning method for relation extraction, *Expert Systems with Applications* **248** (2024), 123478. doi:<https://doi.org/10.1016/j.eswa.2024.123478>. <https://www.sciencedirect.com/science/article/pii/S0957417424003439>.
- [9] A.D. Cohen, S. Rosenman and Y. Goldberg, Relation classification as two-way span-prediction, *arXiv preprint arXiv:2010.04829* (2020).
- [10] W. Zhou and M. Chen, 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)*, Y. He, H. Ji, S. Li, Y. Liu and C.-H. Chang, eds, Association for Computational Linguistics, Online only, 2022, pp. 161–168. <https://aclanthology.org/2022.aacl-short.21>.
- [11] B. Li, W. Ye, J. Zhang and S. Zhang, Reviewing Labels: Label Graph Network with Top-k Prediction Set for Relation Extraction, 2022.
- [12] J. Han, S. Zhao, B. Cheng, S. Ma and W. Lu, Generative Prompt Tuning for Relation Classification, in: *Findings of the Association for Computational Linguistics: EMNLP 2022*, Y. Goldberg, Z. Kozareva and Y. Zhang, eds, Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 2022, pp. 3170–3185. doi:10.18653/v1/2022.findings-emnlp.231. <https://aclanthology.org/2022.findings-emnlp.231/>.
- [13] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel and D. Kiela, Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks, in: *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Curran Associates Inc., Red Hook, NY, USA, 2020. ISBN 9781713829546.
- [14] Y.S. Kai Zhang Bernal Jiménez Gutiérrez, Aligning Instruction Tasks Unlocks Large Language Models as Zero-Shot Relation Extractors, in: *Findings of ACL*, 2023.
- [15] W. Xiong, Y. Song, P. Wang and S. Li, Rationale-Enhanced Language Models are Better Continual Relation Learners, in: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, H. Bouamor, J. Pino and K. Bali, eds, Association for Computational Linguistics, Singapore, 2023, pp. 15489–15497. doi:10.18653/v1/2023.emnlp-main.958. <https://aclanthology.org/2023.emnlp-main.958>.
- [16] N. Mihindukulasooriya, S. Tiwari, C.F. Enguix and K. Lata, Text2KGBench: A Benchmark for Ontology-Driven Knowledge Graph Generation from Text, in: *The Semantic Web – ISWC 2023*, T.R. Payne, V. Presutti, G. Qi, M. Poveda-Villalón, G. Stoilos, L. Hollink, Z. Kaoudi, G. Cheng and J. Li, eds, Springer Nature Switzerland, Cham, 2023, pp. 247–265.
- [17] R. Ahmad, M. Critelli, S. Efeoglu, E. Mancini, C. Ringwald, X. Zhang and A. Merono Penuela, Draw Me Like My Triples: Leveraging Generative AI for Wikidata Image Completion, 2023, The 4th Wikidata Workshop ; Conference date: 07-11-2023. <https://wikidataworkshop.github.io/2023/>.
- [18] S. Hertling and H. Paulheim, OLaLa: Ontology matching with large language models, in: *Proceedings of the 12th Knowledge Capture Conference 2023*, 2023, pp. 131–139.
- [19] Y. Zhang, V. Zhong, D. Chen, G. Angeli and C.D. Manning, Position-aware Attention and Supervised Data Improve Slot Filling, in: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, M. Palmer, R. Hwa and S. Riedel, eds, Association for Computational Linguistics, Copenhagen, Denmark, 2017, pp. 35–45. doi:10.18653/v1/D17-1004. <https://aclanthology.org/D17-1004>.
- [20] C. Alt, A. Gabryszak and L. Hennig, TACRED Revisited: A Thorough Evaluation of the TACRED Relation Extraction Task, in: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, D. Jurafsky, J. Chai, N. Schuster and J. Tetreault, eds, Association for Computational Linguistics, Online, 2020, pp. 1558–1569. doi:10.18653/v1/2020.acl-main.142. <https://aclanthology.org/2020.acl-main.142>.
- [21] G. Stoica, E.A. Platanios and B. Póczos, Re-TACRED: Addressing Shortcomings of the TACRED Dataset, *Proceedings of the AAAI Conference on Artificial Intelligence* **35**(15) (2021), 13843–13850. doi:10.1609/aaai.v35i15.17631. <https://ojs.aaai.org/index.php/AAAI/article/view/17631>.
- [22] I. Hendrickx, S.N. Kim, Z. Kozareva, P. Nakov, D. Ó Séaghdha, S. Padó, M. Pennacchiotti, L. Romano and S. Szpakowicz, SemEval-2010 Task 8: Multi-Way Classification of Semantic Relations between Pairs of Nominals, in: *Proceedings of the 5th International Workshop on Semantic Evaluation*, K. Erk and C. Strapparava, eds, Association for Computational Linguistics, Uppsala, Sweden, 2010, pp. 33–38. <https://aclanthology.org/S10-1006>.
- [23] R. Thoppilan, D.D. Freitas, J. Hall, N. Shazeer, A. Kulshreshtha, H.-T. Cheng, A. Jin, T. Bos, L. Baker, Y. Du, Y. Li, H. Lee, H.S. Zheng, A. Ghafouri, M. Menegali, Y. Huang, M. Krikun, D. Lepikhin, J. Qin, D. Chen, Y. Xu, Z. Chen, A. Roberts, M. Bosma, V. Zhao, Y. Zhou, C.-C. Chang, I. Krivokon, W. Rusch, M. Pickett, P. Srinivasan, L. Man, K. Meier-Hellstern, M.R. Morris, T. Doshi, R.D. Santos, T. Duke, J. Soraker, B. Zevenbergen, V. Prabhakaran, M. Diaz, B. Hutchinson, K. Olson, A. Molina, E. Hoffman-John, J. Lee, L. Aroyo, R. Rajakumar, A. Butryna, M. Lamm, V. Kuzmina, J. Fenton, A. Cohen, R. Bernstein, R. Kurzweil, B. Aguera-Arcas, C. Cui, M. Croak, E. Chi and Q. Le, LaMDA: Language Models for Dialog Applications, 2022. <https://arxiv.org/abs/2201.08239>.
- [24] H.W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, E. Li, X. Wang, M. Dehghani, S. Brahma, A. Webson, S.S. Gu, Z. Dai, M. Suzgun, X. Chen, A. Chowdhery, S. Narang, G. Mishra, A. Yu, V. Zhao, Y. Huang, A. Dai, H. Yu, S. Petrov, E.H. Chi, J. Dean, J. Devlin, A. Roberts, D. Zhou, Q.V. Le and J. Wei, Scaling Instruction-Finetuned Language Models, arXiv, 2022. doi:10.48550/ARXIV.2210.11416. <https://arxiv.org/abs/2210.11416>.

- [25] H. Touvron, L. Martin, K.R. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D.M. Bikel, L. Blecher, C.C. Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A.S. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I.M. Kloumann, A.V. Korenev, P.S. Koura, M.-A. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E.M. Smith, R. Subramanian, X. Tan, B. Tang, R. Taylor, A. Williams, J.X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov and T. Scialom, Llama 2: Open Foundation and Fine-Tuned Chat Models, *ArXiv* **abs/2307.09288** (2023). <https://api.semanticscholar.org/CorpusID:259950998>.
- [26] A.Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D.S. Chaplot, D.d.l. Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier et al., Mistral 7B, *arXiv preprint arXiv:2310.06825* (2023).
- [27] S. Pan, L. Luo, Y. Wang, C. Chen, J. Wang and X. Wu, Unifying large language models and knowledge graphs: A roadmap, *IEEE Transactions on Knowledge and Data Engineering* (2024).
- [28] E. Melz, Enhancing LLM Intelligence with ARM-RAG: Auxiliary Rationale Memory for Retrieval Augmented Generation, *arXiv e-prints* (2023), arXiv:2311.
- [29] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun and H. Wang, Retrieval-augmented generation for large language models: A survey, *arXiv preprint arXiv:2312.10997* (2023).
- [30] O. Ovadia, M. Brief, M. Mishaelli and O. Elisha, Fine-tuning or retrieval? comparing knowledge injection in llms, *arXiv preprint arXiv:2312.05934* (2023).
- [31] N. Reimers and I. Gurevych, 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, pp. 3982–3992.
- [32] K. Abramski, R. Improt, G. Rossetti and M. Stella, The "LLM World of Words" English free association norms generated by large language models, 2024. <https://arxiv.org/abs/2412.01330>.
- [33] E.J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang and W. Chen, LoRA: Low-Rank Adaptation of Large Language Models, 2021. <https://arxiv.org/abs/2106.09685>.
- [34] S. Min, X. Lyu, A. Holtzman, M. Artetxe, M. Lewis, H. Hajishirzi and L. Zettlemoyer, Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?, in: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Y. Goldberg, Z. Kozareva and Y. Zhang, eds, Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 2022, pp. 11048–11064. doi:10.18653/v1/2022.emnlp-main.759. <https://aclanthology.org/2022.emnlp-main.759/>.

Appendix A. Dataset Overview

Table 11

An example of query sentences and responses of RAG4RE when integrated different LLMs. Note that some sentences in the benchmarks may contain special characters, such as semicolons, which are included in the prompt templates.

Type	Prompt	T5 XL	T5 XXL	Llama2-7b	Mistral-7b
Simple Query	Question: What is the relation type between head and tail entities in the following sentence? Sentence: The second half consisted of the traditional round-table analysis by a trio of familiar faces: journalist George Will, political strategist Donna Brazile and economist Paul Krugman, along with Pakistani journalist and Taliban expert Ahmed Rashid, from Madrid. Head entity: Ahmed Rashid. Tail entity: journalist. Relation types: list of relation types output format: relation_type	title	title	spouse	no_relation
RAG4RE	Problem Definition: Relation extraction is to identify the relationship between two entities in as sentence. Question: What is the relation type between head and tail entities according to given relationships below in the following sentence? Example Sentence: But there is also a bench factor: Williams’ success is partly his own work, partly the legacy of his predecessor Tom Brokaw and partly the strength of the network’s best reporters: Lisa Myers, David Gregory, Campbell Brown, Tim Russert and, in Baghdad, Richard Engel. Sentence: The second half consisted of the traditional round-table analysis by a trio of familiar faces: journalist George Will , political strategist Donna Brazile and economist Paul Krugman, along with Pakistani journalist and Taliban expert Ahmed Rashid, from Madrid. Head: Ahmed Rashid. Tail: journalist. Relation types:list of relation types output format: relation_type,	no_relation	occupation	no_relation	event_time

Table 12

Some data from benchmark datasets

Dataset	Sentence	Entities	Relation
TACRED	He has served as a policy aide to the late U.S. Senator Alan Cranston, as National Issues Director for the 2004 presidential campaign of Congressman Dennis Kucinich, as a co-founder of Progressive Democrats of America and as a member of the international policy department at the RAND Corporation think tank before all that.	Head: Progressive Democrats of America, Tail: international policy department	<i>no_relation</i>
SemEval	The <e1>surgeon</e1> cuts a small <e2>hole</e2> in the skull and lifts the edge of the brain to expose the nerve.	e1: surgeon, e2: hole	Product-Producer (e1,e2)

Appendix B. SemEval Dataset Statistics

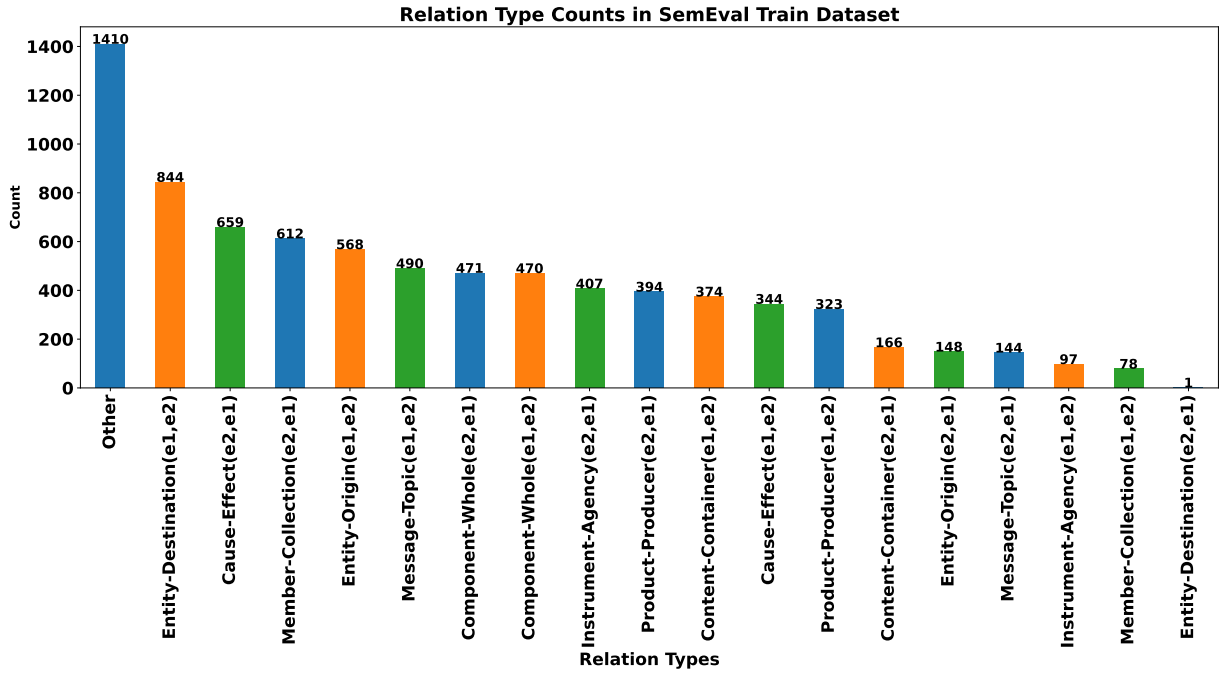


Fig. 6. Details about the number of relation types in SemEval train dataset which is used by Embedding Database in Data Augmentation of RAG4RE.

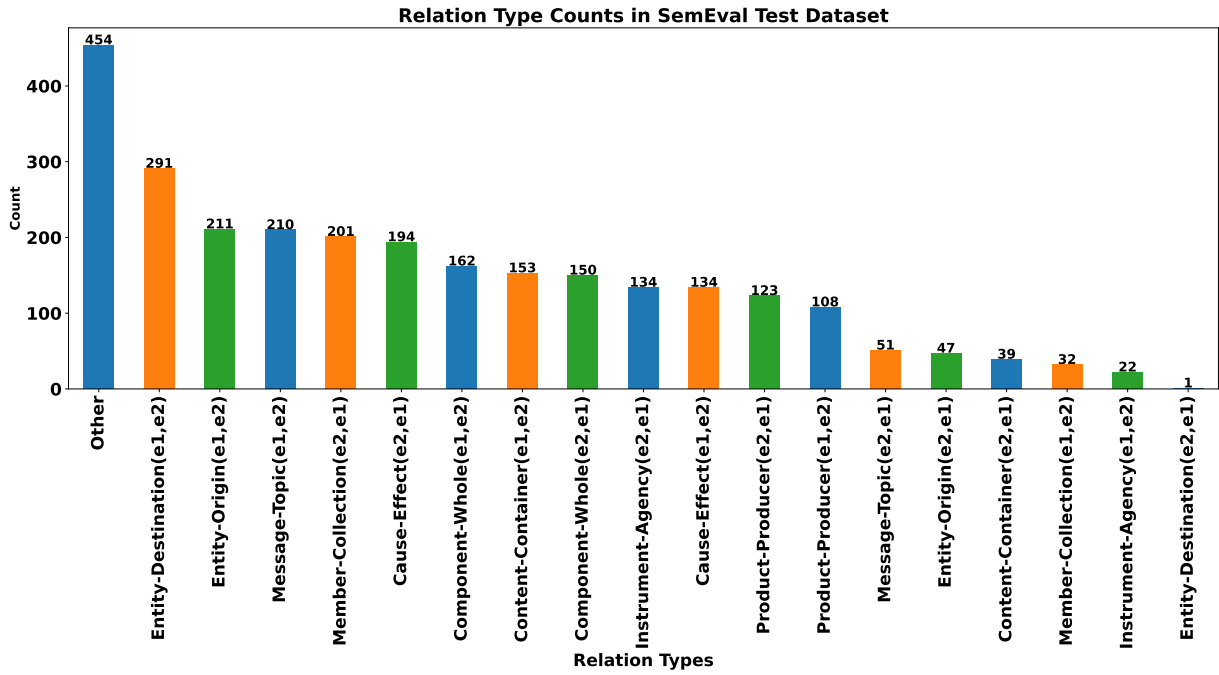
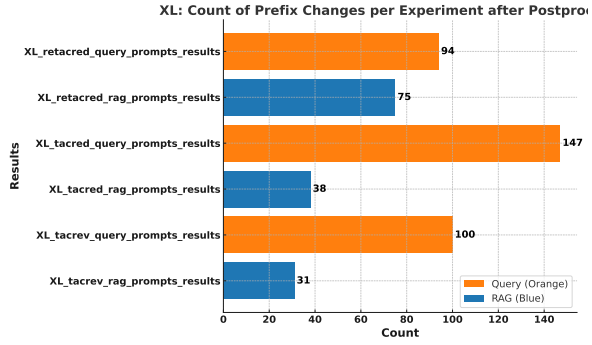
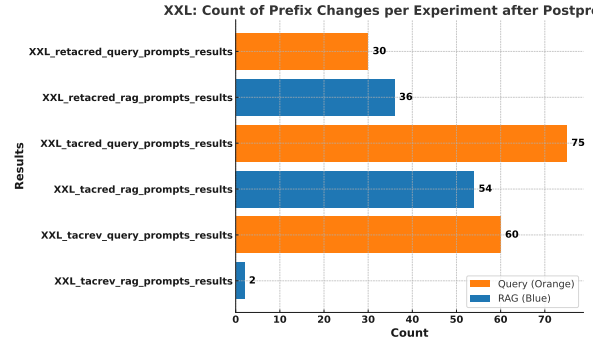


Fig. 7. Details about the number of relation types in SemEval test dataset

Appendix C. Results



((a)) Flan T5 XL prefix changes after post-processing.



((b)) Flan T5 XXL prefix changes after post-processing.

Fig. 8. Prefix changes for Flan T5 models after post-processing.

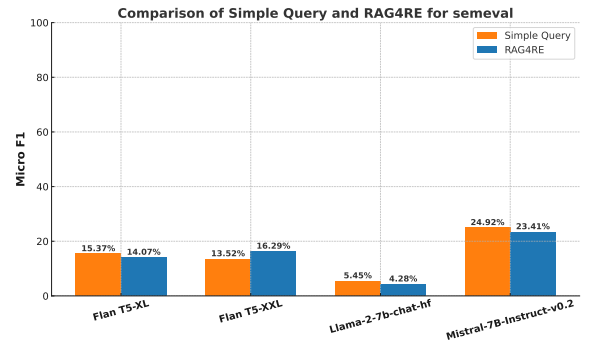
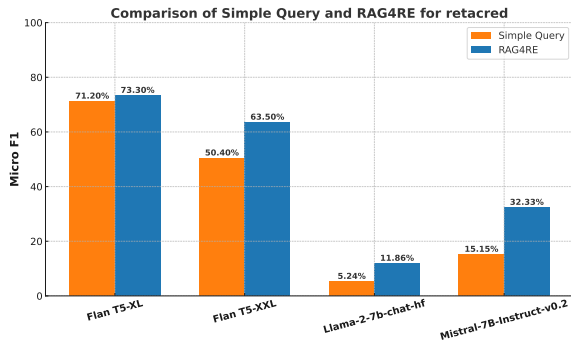
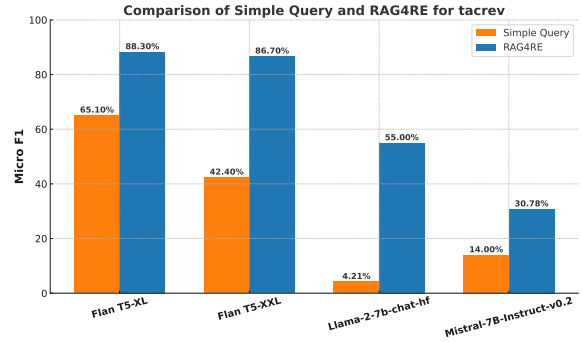
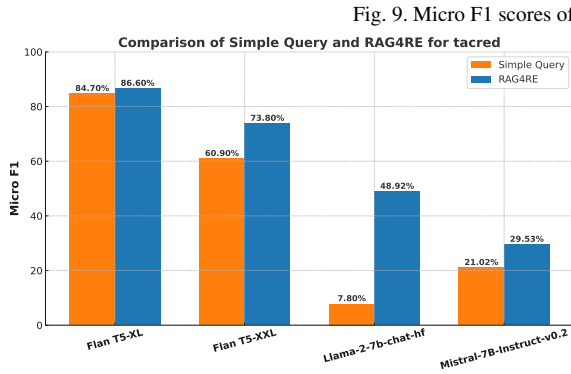


Fig. 9. Micro F1 scores of four different benchmark datasets.

Appendix D. Ablation Study

Sample of One-Shot Prompt Template for RAG4RE

Problem Definition: Relation extraction is to identify the relationship between two entities in as sentence.
Question : What is the relation type between head and tail entities according to given relationships below in the following sentence?
Example Sentence: He is also an adviser to the Senatorial Committee of the US Congress and to the Democratic Party.
Head: He
Tail: US
Relation type: no_relation
Sentence: He has served as a policy aide to the late U.S. Senator Alan Cranston , as National Issues Director for the 2004 presidential campaign of Congressman Dennis Kucinich , as a co-founder of Progressive Democrats of America and as a member of the international policy department at the RAND Corporation think tank before all that .
Head: Progressive Democrats of America.
Tail: international policy department.
Relation types: list of relation types
output format: relation_type

Fig. 10. A sample for one-shot prompt template generated from TACRED dataset.

Sample of Prompt Template for RAG4RE

Example Sentence: He is also an adviser to the Senatorial Committee of the US Congress and to the Democratic Party .
Query Sentence: He has served as a policy aide to the late U.S. Senator Alan Cranston , as National Issues Director for the 2004 presidential campaign of Congressman Dennis Kucinich , as a co-founder of Progressive Democrats of America and as a member of the international policy department at the RAND Corporation think tank before all that .
 What is the relation type between **Progressive Democrats of America** and **international policy department** according to given relation types below in the sentence?
Relation types: list of relation types

Fig. 11. Illustration of a sample for Prompt Template used in RAG4RE in ablation study. The relation type is 'no_relation' between entities.