Data Engineering Research Group
TUM School of Computation, Information and Technology
Technical University of Munich

Evaluating Prompt Engineering Methods for Relation Extraction

A Case Study in the Biomedical Domain

Bachelor Practical Course on Data Engineering (WiSe 2026)

Seymen Yildirim ✉ **and Long San Dennis Lai** ✉

✉ seymen.yildirim@tum.de
✉ dennislai222.lai@tum.de

January 21, 2026

Abstract — The increasing availability of unstructured text, together with the rapid adoption of large language models (LLMs), presents new opportunities for automated relation extraction. However, it remains unclear which prompt engineering strategies, prompt complexity levels, and LLM choices are the most effective for this task. This study evaluates four commonly used prompt engineering methods—input-output prompting (I/O), chain-of-thought prompting (CoT), retrieval-augmented generation (RAG), and reason-and-act prompting (ReAct)—across ten different commercially available LLMs at three prompt complexity levels. The experiments show that reasoning-based techniques, particularly ReAct and CoT, consistently achieve the strongest performance across models and prompt complexity levels. Moreover, prompts of moderate complexity, which incorporate a small number of examples and relation type definitions, yield the best overall results. In addition, LLMs struggle to classify relation types accurately despite being highly capable of locating related entity pairs. These findings act as guidance for designing more effective relation extraction pipelines using LLMs.

1 Introduction

Relation extraction has gained popularity as a technique to extract key information from unstructured text by identifying semantic relationships between entities over the past decade, as demonstrated by the increasing number of relation extraction papers published at major ACL venues between 2020 and 2023 [1].

The introduction of big data, paired with the recent rise in popularity of large language models, has led researchers to question whether the cost of relation extraction can be reduced through automation. Although multiple studies have already explored this direction, new large language models developed by various companies continue to emerge, each exhibiting their own strengths and limitations. As a result, further evaluation on these new models remains necessary. This study therefore focuses on assessing various prompt engineering techniques for relation extraction across multiple large language models.

While relation extraction is applied across multiple domains, this study focuses on the biomedical domain due to its high information density, complex terminology, and the availability of high-quality annotated datasets, such as the BioRED dataset [2].

2 Background

2.1 Definitions

1) Relation Extraction (RE)

Relation extraction refers to the process of identifying and distinguishing semantic relations among entities in text [3], which can be a sentence, a document, or even multiple documents [4].

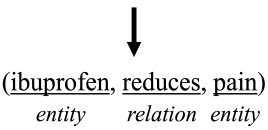

Figure 1. Example of relation extraction from a sentence.

2) Prompt Engineering

Prompt engineering refers to the process of designing, constructing, and refining contextually relevant queries, with the intent of inducing responses from large language models (LLMs) which aid in specific tasks [5] [6].

3) Prompt Engineering Methods

Prompt engineering methods (also referred to as prompting techniques or strategies) are systematic approaches for designing and refining prompts which influence the behavior and outputs of large language models.

a) Input-Output Prompting (I/O)

In input-output prompting, the model receives examples consisting of inputs and their corresponding outputs, which the model learns from to generate appropriate responses for new inputs [7].

b) Chain-of-Thought Prompting (CoT)

In chain-of-thought prompting, the model is encouraged to decompose a problem into a sequence of reasoning steps, which are then presented together with the final solution as output [7].

c) Retrieval-Augmented Generation (RAG)

In retrieval-augmented generation, an external knowledge source is used to retrieve information relevant to the prompt, which is incorporated into the model's generation process to support knowledge-intensive tasks [8].

d) Reason-and-Act Prompting (ReAct)

In reason-and-act prompting, the model is instructed to generate reasoning steps and corresponding actions in an alternating sequence, which allows it to both reason about and act within a task while producing a solution [9].

4) Evaluation Metrics for Relation Extraction

To systematically evaluate the performance of different prompting strategies, five evaluation metrics are introduced. Here, TP, FP, and FN denote true positives, false positives, and false negatives respectively.

a) Precision (P)

Precision denotes the proportion of extracted relations that are relevant, formally defined as:

$$P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{1}$$

b) Recall (R)

Recall measures the proportion of gold standards that were successfully retrieved, formally defined as:

$$R = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{2}$$

c) F1 Score (F1)

The F1 score is the harmonic mean that balances precision and recall, formally defined as:

$$\mathrm{F1} = \frac{2 \times R \times P}{R + P} \tag{3}$$

d) Omission Rate (OR)

The omission rate measures the percentage of relations which are present in the gold standard but not extracted, formally defined as:

$$\mathrm{OR} = \frac{\mathrm{FN}}{\mathrm{TP} + \mathrm{FN}} \tag{4}$$

e) Hallucination Rate (HR)

The hallucination rate determines the proportion of incorrectly retrieved relations, formally defined as:

$$\mathrm{HR} = \frac{\mathrm{FP}}{\mathrm{TP} + \mathrm{FP}} \tag{5}$$

3 Related Work

3.1 Prompt Engineering for Relation Extraction with Large Language Models

To position this study with respect to prior research, two relevant studies were selected for review. Polat et al. compared different prompt engineering methods for knowledge extraction from text [10]. In this study, all four of the aforementioned methods were tested. All experiments were conducted using a single artificial intelligence (AI) model, GPT-4, on texts sourced from Wikipedia and Wikidata on knowledge-intensive domains. Their tests revealed that incorporating a single relevant example into prompts could improve extraction performance two- to threefold. Furthermore, reasoning-based methods, such as CoT and ReAct, did not perform substantially better than simpler prompting methods. Among the evaluated methods, RAG achieved the highest scores in precision, recall, and F1.

While Polat et al. focused on open-domain knowledge extraction, Dong et al. investigated relation extraction in the biomedical domain, similar to the current study [11]. They introduced SyRACT, a prompting strategy which combines RAG and CoT. Relations were extracted from biomedical documents and abstracts obtained from PubMed and evaluated against three existing professional biomedical datasets. It was observed that RAG significantly reduced hallucination by supplying verified domain-specific context, while CoT improved interpretability of relations extracted from longer biomedical documents by encouraging the model to explicitly justify entity interactions. Nevertheless, SyRACT attained consistently higher precision, recall, and F1 scores across the evaluated biomedical datasets, outperforming the individual prompting techniques by combining their strengths.

3.2 Motivation for the Present Study

While the study by Polat et al. [10] was the most closely related to the present work, the prompting techniques were only tested using a single model. Similarly, the study by Dong et al. [11], despite also focusing on biomedical datasets, only considered a restricted set of prompting techniques and did not explicitly specify the LLMs used. To address these limitations, the current study seeks to provide a more comprehensive understanding of the topic by systematically evaluating multiple prompting strategies across multiple LLMs for relation extraction in the biomedical domain.

4 Methodology

4.1 Dataset

In this study, all experiments were conducted using the BioRED dataset [2], which provides high-quality document-level annotations for biomedical relation extraction from PubMed abstracts. In the dataset, eight semantic relation types were defined: *Association*, *Positive_Correlation*, *Negative_Correlation*, *Bind*, *Cotreatment*, *Comparison*, *Drug_Interaction*, and *Conversion*, each representing a different interaction between the entities examined.

To ensure experimental results would be comparable to prior work, the experiments were performed on a subset of 50 documents sampled from the test split of the BioRED dataset.

4.2 Experimental Setup

In this study, four prompting techniques were evaluated: I/O, CoT, RAG, and ReAct. Each prompting technique was implemented at three complexity levels: *baseline*, which provides minimal instructions; *improved*, which includes definitions of the relation types and two to three examples; and *full*, which supplies comprehensive instructions along with six examples.

To evaluate performance across commercially available models, ten LLMs from multiple providers were evaluated in this study: GPT-4o, GPT-4o-mini, GPT-4.1, GPT-5-mini, GPT-5-nano (OpenAI); Claude Sonnet 4.5 (Anthropic); Gemini 2.0 Flash (Google); Llama 3.1 70B (Meta); DeepSeek Chat v3.1; and Mistral Nemo.

By varying the prompting techniques, the complexity levels of prompts, and LLMs, a total of $4 \times 3 \times 10 = 120$ distinct model-prompt configurations were obtained.

4.3 Evaluation

To quantify the performance of the model-prompt combinations, strict exact entity matching was used, which requires predicted entities to completely match the entities in the gold standard as identical strings. Although other matching strategies allow for higher absolute scores, exact matching was chosen as it provides a conservative evaluation which enables a more fine-grained comparison of performance across the model-prompt combinations.

Two evaluation modes were considered for strict exact entity matching: *with-types*, which requires both the entity pair and relation type to match the gold standard; and *entity-only*, which ignores relation types and only measures if the correct entity pairs are identified.

Performance was measured using macro-averaged F1, precision, and recall scores, alongside omission and hallucination rates. In addition, an aggregated overall score (OS) is defined:

$$\text{OS} = \text{F1} \times 0.5 + P \times 0.2 + R \times 0.2 + (1 - \text{OR}) \times 0.1 \quad (6)$$

4.4 Reproducibility

For reproducibility, the complete experimental pipeline used in this study is publicly available as a GitHub repository[1].

5 Results

5.1 Comparison of Prompting Techniques

Figure 2 shows the performance of each prompting technique across all evaluated models. The heatmap reveals a clear pattern across the three prompt-complexity groups: while differences in performance were minimal at the baseline level, ReAct and CoT consistently scored higher than I/O and RAG at the improved and full complexity levels across most models.

Figure 3 presents the macro-averaged F1, precision, and recall scores for each technique-complexity combination, averaged across all evaluated models. At the baseline complexity level, only negligible differences were observed across techniques. At the improved and full complexity levels, differences in F1 and precision remained relatively small; however, ReAct and CoT produced noticeably higher recall scores.

5.2 Best Technique by Model

Figure 2 also showcases the technique-complexity configuration that achieved the highest performance for each evaluated model, indicated in bold. Across all models, the improved prompt complexity level obtained the best results. Among prompting methods, ReAct achieved the highest F1 score in general and performed best for Claude Sonnet 4.5, while CoT attained the best performance for several other models.

5.3 Impact of Prompt Complexity

Figure 2 reports differences in performance across the three prompt complexity levels. Prompts at the improved level attained F1 scores ranging from 0.161 to 0.372, outperforming baseline prompts (0.034-0.150) and generally exceeding full prompts (0.116-0.316) across models and techniques.

5.4 Entity Identification vs. Relation Type Classification

Figure 4 shows a substantial performance gap between entity-only and with-types evaluation. Overall scores, as defined in section 4, were higher for entity-only evaluation than for with-types evaluation across all models. As a representative example, with-types evaluation with Claude Sonnet 4.5 achieved an overall score of 0.263, compared to 0.497 under entity-only evaluation, which corresponds to a relative improvement of 89%.

5.5 Best Configuration

Under strict exact entity matching, the best overall configuration was improved-ReAct with Claude Sonnet 4.5, as shown in Figure 2.

6 Discussion

6.1 Interpretation of Results

ReAct achieved the best overall performance among the evaluated technique-complexity configurations, closely followed by CoT (see Figure 2). Although ReAct and CoT are closely related, the former allows models to alternate between reasoning, producing, and modifying preliminary results until a solid conclusion is reached, while the latter requires models to state all reasoning steps before producing a final answer. This difference in flexibility most likely contributed to the better performance of ReAct. In addition, ReAct performed particularly well on larger-capacity models, such as Claude Sonnet 4.5, Llama 3.1 70B, and Gemini 2.0 Flash (see Figure 2), suggesting that reasoning-action workflows benefit from increased model capacities.

It is further observed that prompt structure had a substantial impact on model performance (see Figure 3). While prompts at the baseline complexity level provided insufficient guidance for executing tasks, prompts at the full complexity level, on the contrary, overwhelmed the models with excessive detail. Improved prompts, on the other hand, offered a balanced middle ground, providing sufficient guidance whilst allowing models flexibility in their reasoning process.

Furthermore, it is noted that the LLMs were significantly better at identifying related entity pairs than at correctly

[1] https://github.com/Soeky/relation-extraction-using-llms

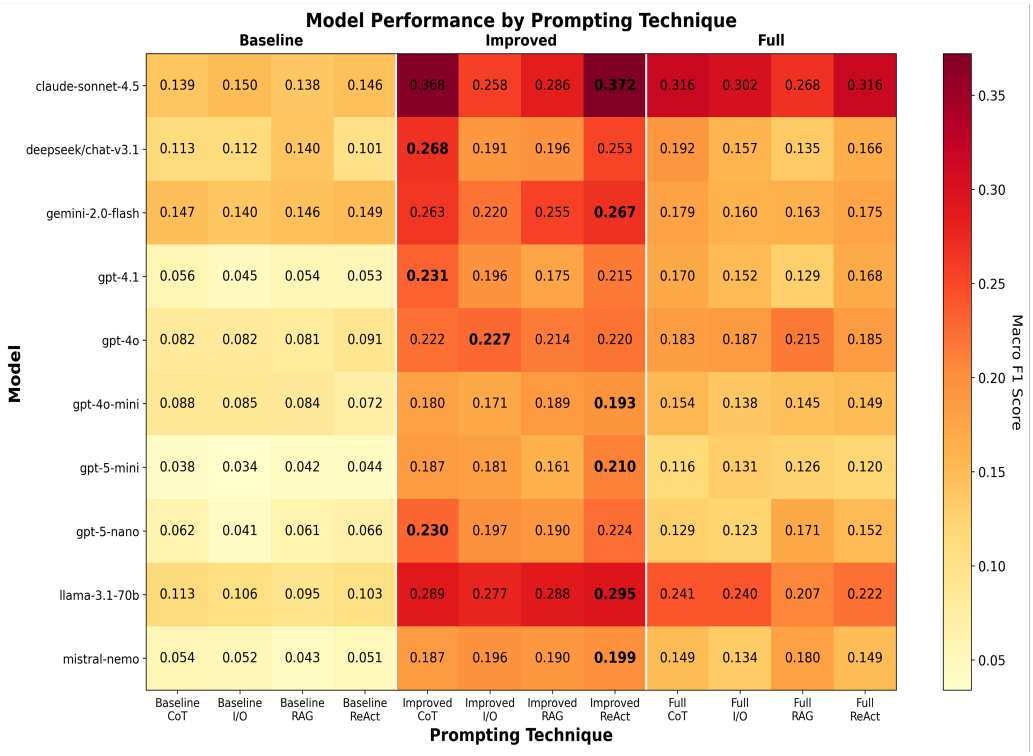

Figure 2. F1 scores across prompting techniques and models, grouped by prompt complexity level. Darker colors indicate higher performance.

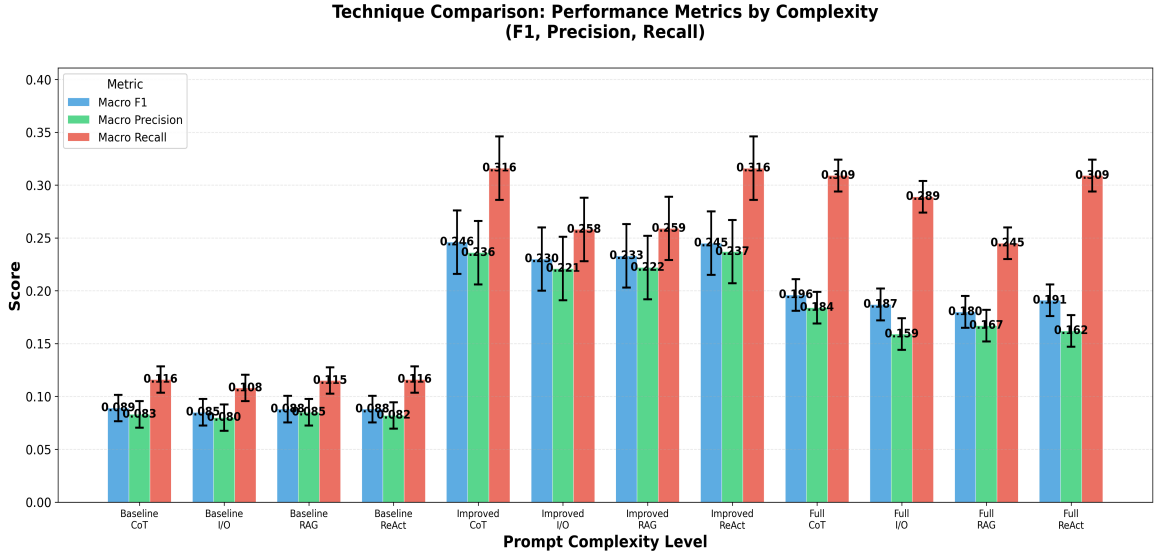

Figure 3. Comparison of prompting techniques and complexities across Precision, Recall, and F1 metrics, with standard deviation.

classifying their relation types (see Figure 4). Despite often succeeding in identifying related entities, the models frequently misclassified semantically similar relation types such as *Positive_Correlation* and *Association*, or *Negative_Correlation* and *Drug_Interaction*. This is presumably because entity identification can be considered a prerequisite subtask of relation type classification, and suggests that classifying relation types remains a challenge for models which are not fine-tuned for the task.

Nevertheless, the results show that relation extraction using LLMs still leaves room for improvement. The average omission rate across all model-prompt configurations was 65%, which suggests that a significant proportion of gold-standard relations were not extracted. These omissions commonly involved relations which spanned multiple sentences or were not explicitly stated and required implicit inference. Besides, the average hallucination rate reached 70%, which implies that models often generated plausible relations not present in the texts, possibly based on external domain knowledge.

6.2 Comparison with Prior Work

The results of this study contradict the findings from Polat et al. [10], showing that ReAct and CoT achieve stronger

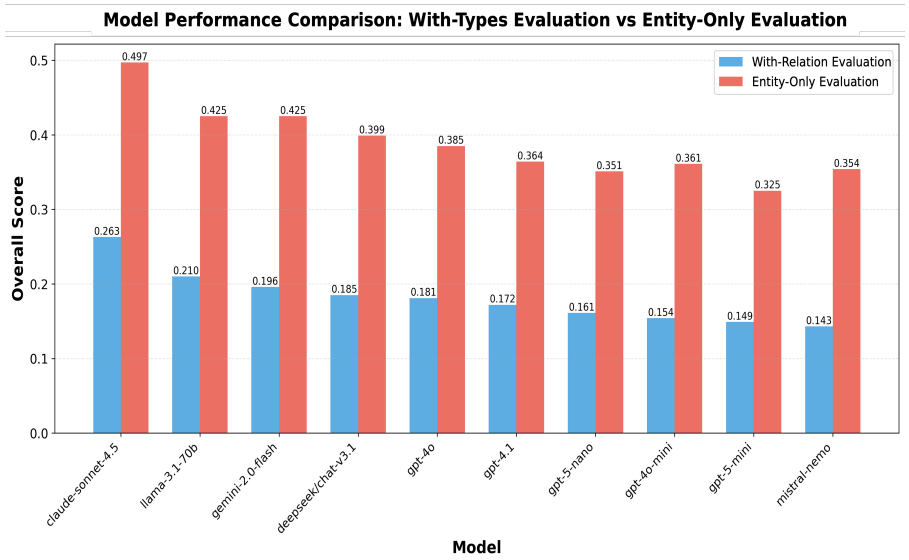

Figure 4. Performance comparison between entity-only and with-types evaluation. Scores are averaged across all prompting techniques and complexities.

performance than other techniques. This discrepancy could be due to differences in experimental settings between the two studies. At the same time, this study concurs with Polat et al. in observing that adding a small number of relevant examples boosts extraction performance, while too many examples lead to diminishing returns.

While Dong et al. reported SyRACT as the best-performing approach [11], this study finds that ReAct and CoT individually achieve the strongest performance across a broader range of models. In contrast to Dong et al., where RAG substantially reduces hallucination rates, this study shows that RAG exhibits high variability across different models and prompt complexity levels, suggesting that it is more sensitive to the underlying model and prompt formulation than reasoning-based techniques.

6.3 Limitations and Threats to Validity

Firstly, experiments were conducted on a subset of only 50 documents from the BioRED test split due to computational constraints. As a result, the findings may not fully capture the variability present in the complete BioRED test set.

Secondly, all experiments were limited to the biomedical domain. The performance of relation extraction methods may differ when applied to other knowledge-intensive domains, such as the legal and financial domains, each having their own domain-specific language and relations.

Lastly, the primary results of this study were based solely on strict exact entity matching, which may underestimate performance when models generate outputs with different wording but the same meaning. Although further matching strategies exist, they were not included in the main results to allow for a conservative and consistent evaluation.

6.4 Future Work

The substantial performance gap between entity-only and with-types evaluation opens up possibilities for future research. A multi-stage relation extraction pipeline could be explored, in which separate models are used for locating entity pairs and assigning relation types. This approach would allow for each subtask to be optimized independently, thus reducing classification errors.

7 Conclusion

In this study, four commonly used prompt engineering methods for relation extraction—namely I/O, CoT, RAG, and ReAct—were evaluated across ten commercially available large language models. Each model-technique combination was further tested at three different prompt complexity levels which varied in the amount of instruction detail and the number of examples provided. The results indicate that ReAct and CoT consistently achieved the strongest performance across models, while including relation type definitions and a small number of examples in prompts led to the best overall results across all prompting techniques and models.

Beyond prompt design, choosing the correct model was also discovered to play a critical role in relation extraction performance. The results show that model selection heavily influenced the performance of relation extraction tasks, regardless of the prompting strategy employed. In particular, larger models typically performed better than smaller ones, likely because they are trained on more data and are more capable of relation extraction.

As relation extraction continues to gain importance with the growing availability of textual data and the widespread adoption of large language models, its potential for real-world applications is expected to increase. By systematically analyzing the effects of prompt engineering strategies, prompt complexity, and model choice, this study provides practical insights which can inform the design of more effective relation extraction pipelines in future applications.

References

[1] J. Diaz-Garcia and J. Lopez, "A survey on cutting-edge relation extraction techniques based on language models," *Artificial Intelligence Review*, vol. 58, 07 2025.

[2] L. Luo, P.-T. Lai, C.-H. Wei, C. N. Arighi, and Z. Lu, "Biored: a rich biomedical relation extraction dataset," *Briefings in Bioinformatics*, vol. 23, 07 2022.

[3] Y. S. Chan and D. Roth, "Exploiting background knowledge for relation extraction," in *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)* (C.-R. Huang and D. Jurafsky, eds.), (Beijing, China), pp. 152–160, Coling 2010 Organizing Committee, Aug. 2010.

[4] M. Jain, "Knowledge enabled relation extraction," in *Companion Proceedings of the ACM Web Conference 2024*, WWW '24, (New York, NY, USA), p. 1210–1213, Association for Computing Machinery, 2024.

[5] A. Bozkurt and R. C. Sharma, "Generative ai and prompt engineering: The art of whispering to let the genie out of the algorithmic world," *Asian Journal of Distance Education*, vol. 18, July 2023.

[6] B. Meskó, "Prompt engineering as an important emerging skill for medical professionals: Tutorial," *J Med Internet Res*, vol. 25, p. e50638, Oct 2023.

[7] M. Abedi, I. Alshybani, M. Shahadat, and M. Murillo, "Beyond traditional teaching: The potential of large language models and chatbots in graduate engineering education," Sept. 2023.

[8] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. tau Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-augmented generation for knowledge-intensive nlp tasks," 2021.

[9] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao, "React: Synergizing reasoning and acting in language models," 2023.

[10] F. Polat, I. Tiddi, and P. Groth, "Testing prompt engineering methods for knowledge extraction from text," *Semantic Web*, vol. 16, no. 2, pp. SW–243719, 2025.

[11] X. Dong, D. Zhao, J. Meng, B. Guo, and H. Lin, "Syract: zero-shot biomedical document-level relation extraction with synergistic rag and cot," *Bioinformatics*, vol. 41, p. btaf356, 06 2025.