Improving Automatic Evaluation of Large Language Models (LLMs) in Biomedical Relation Extraction via LLMs-as-the-Judge

Anonymous ACL submission

Abstract

Large Language Models (LLMs) have demonstrated impressive performance in biomedical relation extraction, even in zero-shot scenarios. However, evaluating LLMs in this task remains challenging due to their ability to generate human-like text, often producing synonyms or abbreviations of gold-standard answers, making traditional automatic evaluation metrics unreliable. On the other hand, while human evaluation is more reliable, it is costly and time-consuming, making it impractical for real-world applications. This paper investigates the use of LLMs-as-the-Judge as an alternative evaluation method for biomedical relation extraction. We benchmark 8 LLMs as judges to evaluate the responses generated by 5 other LLMs across 3 biomedical relation extraction datasets. Unlike other text-generation tasks, we observe that LLM-based judges perform quite poorly (usually below 50% accuracy) in the biomedical relation extraction task. Our findings reveal that it happens mainly because relations extracted by LLMs do not adhere to any standard format. To address this, we propose structured output formatting for LLM-generated responses that helps *LLM-Judges* to improve their performance by about 15% (on average). We also introduce a domain adaptation technique to further enhance *LLM-Judge* performance by effectively transferring knowledge between datasets. We will release both our human-annotated and LLM-annotated judgment data (36k samples in total) for public use.

1 Introduction

Relation extraction, the task of identifying meaningful associations between biomedical entities such as drugs, diseases, and genes from vast amounts of unstructured text (Bassignana and Plank, 2022), is a cornerstone of biomedicine. As the volume of unstructured biomedical text grows exponentially, efficient and accurate relation extraction is no longer a convenience but a necessity for advancing medical research, drug discovery, and improving patient outcomes (Luo et al., 2022). Recent research has shown that Large Language Models (LLMs) can achieve strong performance in biomedical relation extraction, even in zero-shot scenarios (Jahan et al., 2024). This capability makes LLMs particularly valuable for real-world biomedical applications where manually annotated datasets are scarce or costly to obtain.

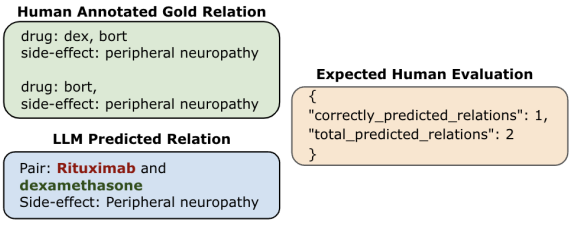

Figure 1: An example of human evaluation of LLM-generated outputs in comparison to the gold labels. The drug "dex" in the gold relation is predicted as "dexamethasone" in LLM response. While exact match will fail in this case, human evaluators can recognize.

However, LLM's ability to generate human-like text introduces a major challenge in evaluation (Laskar et al., 2024a; Jahan et al., 2024). Traditional automatic evaluation methods, such as string matching and token overlap, fail to capture semantic equivalence, as LLMs frequently produce synonyms, abbreviations, or paraphrased responses that are meaningfully correct but not exact matches. For example, an LLM may generate "Hepatic carcinoma" instead of "Liver cancer", leading conventional metrics to misclassify correct extractions as incorrect. Due to these limitations, human evaluation has been the predominant method (see Figure 1 for an example) for assessing LLM performance in biomedical relation extraction (Jahan et al., 2024). However, human evaluation is slow and costly, making it impractical for real-world applications.

To address this, LLMs-as-the-Judge (Zheng et al., 2023) have gained attention as a poten-

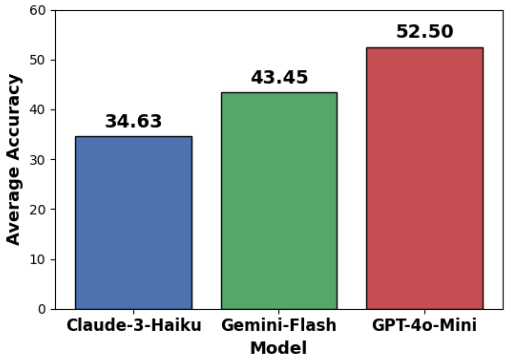

Figure 2: Average judgment accuracy based on our evaluation of different LLMs-as-the-judge across 3 Relation Extraction datasets (KD-DTI, DDI, BC5CDR) to evaluate the LLM-generated responses in Jahan et al. (2024).

tial alternative to human evaluation. While this paradigm has been explored in general NLP tasks (Li et al., 2024a; Gu et al., 2024), there is currently no biomedical-specific *LLM-Judge* designed to evaluate relation extraction tasks. Unlike open-domain text generation, biomedical relation extraction requires precise domain knowledge, standardized terminology, and strict adherence to extract relationships between entities. This complexity raises concerns about whether existing *LLM-Judges* can reliably assess biomedical extractions.

To investigate this, we examined the capability of several LLMs-as-Judges in evaluating the responses generated by different LLMs (*LLM-Generators*) across multiple biomedical relation extraction datasets (Jahan et al., 2024). Surprisingly, despite prior success in general NLP evaluation tasks, *LLM-Judges* performed very poorly in biomedical relation extraction in comparison to human evaluators (Figure 2). These findings suggest that domain specificity may significantly impact the effectiveness of LLMs-as-Judge, underscoring the need for adaptation in biomedicine.

To address the shortcomings of LLMs-as-Judge in the biomedical relation task, we propose *structured output format* (Xia et al., 2024; Li et al., 2024b) for response generation by *LLM-Generators*. Based on extensive experiments, we find that structured output format in the response generated by *LLM-Generators* consistently helps the *LLM-Judges* to improve their performance across various relation extraction datasets.

Moreover, we also find that there is a lack of human-annotated judgment data that prohibits the training of *LLM-Judges* for relation extraction. Therefore, we propose a domain adaptation (Laskar et al., 2022) technique to address the lack of human-annotated judgment data by effectively transferring knowledge from out-of-domain data to improve the performance of *LLM-Judges* in the target domain. Our major contributions are summarized below:

(i) We provide the first comprehensive study of *LLM-Judges* in biomedical relation extraction, benchmarking 8 *LLM-Judges* on responses generated by 5 *LLM-Generators* across 3 biomedical relation extraction datasets. Our findings demonstrate that LLMs are not reliable to serve as evaluators in biomedicine, highlighting their significant performance gap compared to human evaluators.

(ii) To address the above limitation, we propose structured output formatting in LLM-generated responses for biomedical relation extraction to improve the performance of *LLM-Judge*.

(iii) We also propose a domain adaptation technique to effectively transfer knowledge from one domain to another to further improve *LLM-Judge* performance in biomedical relation extraction.

(iv) We conduct over 100 experiments, analyzing the impact of structured output format, domain adaptation, and model scaling on *LLM-Judge* performance. These experiments reveal critical insights into why LLM-based evaluation fails in biomedical relation extraction, establishing the need for task-specific evaluation frameworks. We will make our judgment data (4k human-annotated and 32k LLM-annotated samples) consisting of 3 relation extraction datasets publicly available.

2 Background

2.1 Relation Extraction Task Description

The relation extraction task aims to extract relations between named entities in a given text (Zhong and Chen, 2021). The biomedical relation extraction task aims to identify relationships between biomedical named entities like genes, drugs, and diseases (Chen et al., 2023). More specifically, in the context of biomedicine, the goal is to analyze textual data to identify which gene/variant is responsible or which treatment/drug is effective for which disease, as well as identifying drug-drug interactions, and etc. An example for disease-treatment relation extraction is given below.

Example Text: *The patient has been given chemotherapy for their rare form of cancer.*

Expected Relation: *"Chemotherapy" is a treatment for "rare form of cancer".*

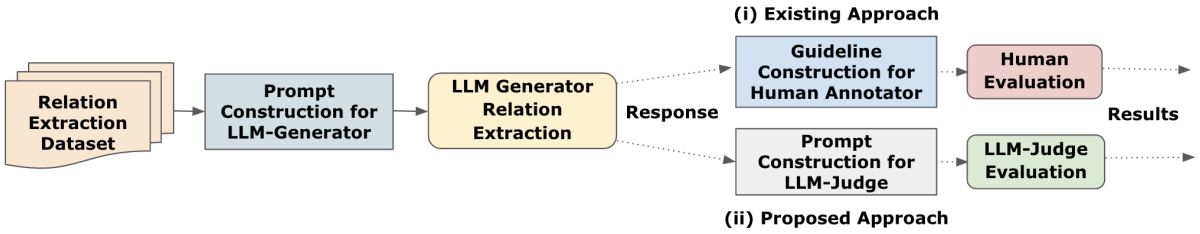

Figure 3: An Overview of LLM-based Relation Extraction System. After constructing the input prompt for the *LLM-Generator* for a given dataset, the *LLM-Generator* generates a response. The evaluation requires the identification of ***the number of correctly predicted relations*** and ***the total number of predicted relations*** in the response to evaluate the performance of the *LLM-Generator* for relation extraction in terms of metrics like Precision, Recall, and F1.

2.2 Related Work

Traditional approaches for biomedical relation extraction relied on supervised learning techniques that required large, manually annotated datasets (Luo et al., 2022). However, the construction of such datasets is expensive and time-consuming. Recently, LLMs have demonstrated impressive zero-shot performance across a wide range of NLP tasks (Laskar et al., 2023; Bang et al., 2023; Qin et al., 2023). This has inspired researchers to successfully apply LLMs for biomedical relation extraction, particularly in zero-shot scenarios where annotated data is scarce (Jahan et al., 2023, 2024; Tian et al., 2024). Despite LLMs achieving impressive zero-shot performance in the biomedical relation extraction task, evaluating the performance of LLM in this task remains challenging (Jahan et al., 2024). This is because LLMs may generate valid outputs that are semantically equivalent to the gold standard but differ in surface form (e.g., synonyms, abbreviations). This issue is further exacerbated in biomedicine due to the nuanced and complex nature of biomedical language. Therefore, existing automatic metrics like string matching or n-gram overlap often fall short in assessing the semantic correctness of LLM-generated free-form responses (Laskar et al., 2024a).

In the context of biomedical relation extraction, while human experts can assess the relevance of LLM extracted relations (Li et al., 2023; Jahan et al., 2024), human evaluation is inherently time-consuming, expensive, and lacks scalability. This makes it impractical for rapid iteration cycles in research and real-world deployment scenarios. The paradigm of using LLMs-as-judges (Zheng et al., 2023) for evaluating free-form text generated by other LLMs has recently gained a lot of attention as a potential alternative to human evaluations (Laskar et al., 2024a). While prior research suggests that LLM-based evaluators can capture linguistic nuances often overlooked by traditional metrics (Li et al., 2024a; Gu et al., 2024), most of these studies have focused on general-domain tasks, without exploring their effectiveness in specialized domains such as biomedical relation extraction. To this end, this paper explores the potential of LLMs-as-the-Judge for evaluating the response generated by other LLMs in the biomedical relation extraction task to mitigate the dependence on expensive, time-consuming human evaluation, alongside ensuring reliability in LLM evaluation by capturing domain-specific nuances that traditional metrics may overlook.

3 Methodology

3.1 LLM-based Relation Extraction Systems

We show the overall pipeline to develop an LLM-based end-to-end relation extraction system in Figure 3. Here, at first, for a given relation extraction dataset, pre-processing steps are first applied to construct a prompt. This prompt is then provided to an LLM (i.e., *LLM-Generator*), which extracts relations. Afterwards, we demonstrate two evaluation paradigms which we describe below:

(i) ***Existing Approaches based on Human Evaluation:*** Where an annotation guideline is first constructed to evaluate the performance of the *LLM-Generator* (i.e., relation extraction LLM) by human annotators. The LLM-generated response is then sent to the human annotators for evaluation.

(ii) ***Our Proposed Approach based on LLM-Judge:*** Where the prompt for the *LLM-Judge* is first constructed to evaluate the performance of the *LLM-Generator*. The response generated by the *LLM-Generator* is then evaluated by the *LLM-Judge*. This is our proposed approach where human evaluator(s) are replaced with the *LLM-Judge*.

In the following, we describe our proposed *LLM-Judge* to evaluate relation extraction models.

3.2 LLM-Judge for LLM-based Relation Extraction Model Evaluation

Prompting: To utilize *LLM-Judge* for the evaluation of LLM-generated responses (i.e., *LLM-Generators*) in relation extraction, we first design the prompt for the judge, as described below.

Prompt to the LLM-Judge

You are required to annotate the response generated by an AI model for biomedical relation extraction. You are given the **relation extraction task description**, followed by the **biomedical text**, then the **human annotated gold relations**, and finally the **AI model predicted relations**. Now, identify how many of the AI predicted relations are correct in comparison to the gold relations. Also, identify how many relations in total are predicted by AI. Generate your response in the JSON format with the following keys:

1. `correctly_predicted_relations`
2. `total_predicted_relations`

[TASK DESCRIPTION]
[BIOMEDICAL TEXT]
[GOLD RELATIONS]
[AI PREDICTED RELATIONS]

We designed this prompt based on extensive prompt engineering using several LLMs (e.g., ChatGPT, Gemini, Claude, etc.) on the outputs generated by different *LLM-Generators* in Jahan et al. (2024) across various relation extraction datasets.

Structured Output Formatting: During prompt engineering in our previous step, we notice that *LLM-Judges* often find it difficult to properly understand the response generated by various *LLM-Generators* since the generated responses are mostly unstructured text. To address this issue, we propose *Structured Output Formatting* for the *LLM-Generators*. In our proposed *Structured Output Formatting* approach, we require the *LLM-Generators* to generate their response in a structured format, i.e., JSON format (see Figure 4 for an example). JSON format was selected since recent research has demonstrated that LLMs are more reliable in generating responses in "JSON" instead of "YAML" (Laskar et al., 2024b).

Domain Adaptation via Transfer Learning: Leveraging closed-source LLMs-as-the-judge have several limitations. For instance, closed-source LLMs are continuously updated. Therefore, the release of an updated version of a closed-source model often makes their respective earlier versions obsolete (Biderman et al., 2024; Laskar et al., 2024a). This creates a huge reproducibility concern. While open-source LLMs could be a suitable alternative, they do not have the superior zero-shot ability of their closed-source counterparts (Jahan et al., 2024). Moreover, we find that existing open-source *LLM-Judges* like Prometheus-2 (Kim et al., 2024) are restricted to certain evaluation dimensions (e.g., pairwise ranking or pointwise scoring for specific criteria: helpfulness, factual correctness, etc.) since they are fine-tuned only for such qualitative metrics. This prohibits the customization of the judging criteria based on user needs, making them inapplicable in our case to evaluate the relation extraction performance of other LLMs.

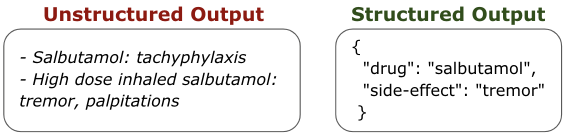

Figure 4: An example of the LLM-generated outputs for *drug* and *drug-induced side-effects* relation extraction in the *unstructured* format used by Jahan et al. (2024) and our proposed *structured* format.

Furthermore, due to the absence of human evaluation data in existing relation extraction datasets for LLM-generated responses, it is difficult to train a reliable *LLM-Judge* model to evaluate the response generated by different relation extraction models. To address this low-resource problem that prohibits the training of *LLM-Judge* for relation extraction, we propose a domain adaptation technique (Garg et al., 2020; Laskar et al., 2022). In our proposed approach, the *LLM-Judge* is fine-tuned on a limited number of human annotated out-of-domain relation extraction data to make it more specialized for relation extraction evaluation. More specifically, suppose we have two datasets, X and Y, where the dataset X may focus on *drug-drug-interaction* extraction and the dataset Y may focus on *disease-treatment-relation* extraction. In our domain adaptation technique, if the target dataset for the judge model to evaluate is the dataset X, then we first fine-tune the judge model on the dataset Y if we have some human evaluation data (e.g., human-annotated judgment labels containing the number of predicted relations and total predicted relations by the *LLM-Generator*) available for the dataset Y. In this way, we specialize the *LLM-Judge* for biomedical relation extraction evaluation.

4 Experiments

To evaluate our proposed solution of replacing human evaluators with *LLM-Judges*, we conduct extensive experiments with various LLMs-as-the-Judge on the responses generated by different *LLM-Generators* for biomedical relation extraction. Below, we first describe the datasets used in our experiments. Then, we describe the LLMs used as the *Judge* and the *Generator*. Afterwards, we present our evaluation metrics to measure the performance of the *LLM-Judge*. Finally, we briefly describe the implementation details.

4.1 Datasets

To evaluate the *LLM-Judges*, we use the responses generated by the *LLM-Generators* for *chemical-disease-relation* extraction (500 test samples) in the BC5CDR dataset (Li et al., 2016), *drug-target-interaction* extraction (1159 test samples) in the KD-DTI dataset (Hou et al., 2022), and *drug-drug-interaction* extraction (191 test samples) in the DDI dataset (Herrero-Zazo et al., 2013).

4.2 Models

LLM-Generators: To evaluate the performance of different *LLM-Judges*, at first we benchmark their performance on different LLM-generated responses collected from Jahan et al. (2024). The responses from the following *LLM-Generators* are used: (i) GPT-3.5 (OpenAI, 2023), (ii) Claude-2 (Anthropic, 2023), (iii) PaLM-2 (Anil et al., 2023), (iv) LLaMA-2-13B (Touvron et al., 2023b). Moreover, to investigate the effectiveness of *Structured Output Format*, we regenerate the response using GPT-4-Turbo (OpenAI, 2023) in both unstructured and structured format, and compare the performance of the *LLM-Judge* when the response is generated in different formats by the same *LLM-Generator*. For this case, we did not use the *LLM-Generators* used in Jahan et al. (2023) since they are early generation LLMs and we observe that they fail to generate the response in structured format.

LLM-Judges: For the *LLM-Judges*, we select the cheapest versions of the currently available closed-source LLMs (GPT-4, Gemini, and Claude-3) to minimize their usage cost. For the open-source *LLM-Judges* (LLaMA, Phi, Qwen, and DeepSeek), we select the models with less than 10B parameters since they can be used in a machine with just 1 L4 GPU (Fu et al., 2024), making them cost-effective for real-world deployment. More specifically, for the *LLM-Judge*, we primarily use the following LLMs: (i) GPT-4o-Mini (OpenAI, 2023), (ii) Gemini-1.5-Flash (Team et al., 2023), (iii) Claude-3-Haiku (Anthropic, 2024), (iv) LLaMA-3.1-8B-Instruct[1] (Dubey et al., 2024), (v) Qwen-2.5-7B-Instruct[2] (Yang et al., 2024b), and (vi) Phi-3.5-Mini-3.8B-Instruct[3] (Abdin et al., 2024). With the recent success of reasoning-based LLMs like DeepSeek-R1 (Guo et al., 2025), we also use its distilled versions based on Qwen and LLaMA, (vii) DeepSeek-R1-Distill-Qwen-7B[4] and (viii) DeepSeek-R1-Distill-Llama-8B[5], respectively. Appendix A.2 contains detailed model descriptions.

While we also tried the biomedical domain-focused BioMistral-7B model (Labrak et al., 2024) as a judge, we observed that it failed to follow the judging instruction and could not evaluate any *LLM-Generators* response (see Appendix A.4.1 for details). Therefore, we did not report its results.

4.3 Evaluation Metrics

We use the following metrics to evaluate the effectiveness of the *LLM-Judge*.

Exact Match Accuracy: Measures the Exact Match Accuracy of the *LLM-Judge* annotation in comparison to the human-annotated gold label. For *exact match* calculation, if there is a match for both the number of *correctly predicted relations* and *total predicted relations* between the gold human annotation and the *LLM-Judge* annotation, then we consider it as an exact match, and the score will be 1. Otherwise, the score will be 0. Thus, a higher exact match score denotes better performance.

Root Mean Squared Error: Measures the Root Mean Squared Error (RMSE) distance between the *LLM-Judge* annotation and the human-annotated gold label. Mean Squared Error (MSE) is defined as the average of the squared differences between the predictions (in this case, the *LLM-Judge* annotations) and the actual values (the *Human-Judge* annotated gold labels). RMSE, which is the square root of the MSE, helps penalize larger errors more

[1] https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct
[2] https://huggingface.co/Qwen/Qwen2.5-7B-Instruct
[3] https://huggingface.co/microsoft/Phi-3.5-mini-instruct
[4] https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Qwen-7B
[5] https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Llama-8B

severely while being in the same units as the target variable. Suppose the *correctly predicted relations* and the *total predicted relations* annotated by humans are 0 and 2, respectively, and by *LLM-Judge* are 1 and 2, respectively, then the RMSE distance:

$$\sqrt{\frac{1}{2}\left((1-0)^2+(2-2)^2\right)} = \sqrt{0.5} \approx 0.71.$$

Contrary to the exact match score, a lower RMSE distance denotes better performance.

4.4 Implementation

For the inference of *LLM-Generators* and *LLM-Judges*, we use the temperature value of 1.0, with other decoding parameters being set to the default values: as given in the respective API providers (OpenAI[6], Google[7], Anthropic[8]) for the closed-source models, and in HuggingFace (Wolf et al., 2020) for the open-source models. Since both the *LLM-Generators* and the *LLM-Judges* do not need to produce outputs of longer sequence length, the maximum output tokens was set to 300 tokens except for the reasoning models (DeepSeek-R1). For the DeepSeek-R1-based models, we increased the output token limit to 1000 tokens to allow the model enough tokens for thinking. For domain adaptation via transfer learning, we fine-tune for 3 *epochs* with the *batch size* = 1, *learning rate* = $2e-5$, and *sequence length* = $3k$. All experiments were run using 1 A100 GPU.

5 Results and Discussion

5.1 Performance in Existing LLM-based Relation Extraction Benchmarks

In this section, we benchmark the performance of different *LLM-Judges* to evaluate the responses generated by different *LLM-Generators* used by Jahan et al. (2024). In their work, Jahan et al. (2024) used GPT-3.5 (OpenAI, 2023), PaLM-2 (Anil et al., 2023), Claude-2 (Anthropic, 2023), and LLaMA-2-13B-Instruct (Touvron et al., 2023b) as the *LLM-Generators*. We collected their human evaluator annotated labels consisting of the number of *correctly predicted relations* and *total predicted relations* for each *LLM-Generator*. Then we measure the Exact Match Accuracy and the RMSE Distance between the *LLM-Judge* annotation (the number of *correctly predicted relations* and the *total predicted relations* annotated by the *LLM-Judge*) and the human annotation.

For each dataset, we then compute the average Exact Match Accuracy and RMSE Distance across all *LLM-Generators*, as demonstrated in Table 1. From Table 1, we find that none of the *LLM-Judges* could reach accuracy above 50%, except GPT-4o-Mini. Nonetheless, GPT-4o-Mini still fails to achieve more than 60% accuracy in any datasets. Our hypothesis is that the responses generated by the *LLM-Generators* used by Jahan et al. (2024) were quite unstructured, which could be difficult for the *LLM-Judges* to evaluate. In the following section, we investigate whether *Structured Output Formatting* could alleviate this issue.

5.2 Performance of LLM-Judges based on Structured vs Unstructured Output

To improve the *LLM-Judge* accuracy, we generate the responses in all three datasets using the GPT-4-Turbo (OpenAI, 2023) model as the generator, with specifically prompting GPT-4-Turbo to extract the relations between the named entities in a more structured way, i.e., JSON format. To compare whether this structured output format could improve the performance of the *LLM-Judges*, we also regenerate the responses without any structured output format instruction by using the same prompt as used by Jahan et al. (2024). Afterward, two human annotators having backgrounds in NLP and biomedicine annotated the *correctly predicted relations* and the *total predicted relations* in the GPT-4-Turbo generated responses for both structured and unstructured cases. When discrepancies arise between annotations from different annotators, they are resolved through discussions.

We then evaluate the performance of *LLM-Judges* in these structured responses as well as the unstructured responses generated by GPT-4-Turbo. Based on the results presented in Table 2, we find that structured formatting consistently improves the performance for all models. While in the BC5CDR dataset, Gemini-1.5-Flash achieves the best performance in terms of accuracy with Claude-3-Haiku achieving the best result in terms of the RMSE metric, in other two datasets (DDI and KD-DTI), GPT-4o-Mini achieves the best performance in terms of both metrics. For open-source LLMs, in the BC5CDR dataset, Qwen-2.5-7B has the best accuracy and LLaMA-3.1-8B performs the best in terms of RMSE distance. However, the

[6] https://platform.openai.com/docs/api-reference/introduction

[7] https://ai.google.dev/gemini-api/docs

[8] https://www.anthropic.com/api

LLM-Judge	BC5CDR		DDI		KD-DTI	
	EM (△)	RMSE (▽)	EM (△)	RMSE (▽)	EM (△)	RMSE (▽)
Phi-3.5-Mini-3.8B-Instruct	33.80	2.40	43.06	2.19	35.55	2.11
Qwen-2.5-7B-Instruct	45.25	2.42	46.60	2.15	49.98	1.82
LLaMA-3.1-8B-Instruct	29.45	2.40	29.32	2.95	36.73	2.10
Deepseek-R1-Qwen-7B	30.60	2.76	42.67	3.07	42.45	2.51
Deepseek-R1-LLaMA-8B	30.50	3.37	42.15	4.16	33.48	3.25
Claude-3-Haiku	29.50	2.26	31.15	2.70	40.27	1.83
Gemini-Flash	42.55	**2.09**	47.12	2.11	40.68	1.98
GPT-4o-Mini	**48.35**	2.33	**59.03**	**1.84**	**53.11**	**1.81**

Table 1: Performance of different *LLM-Judges* on the responses generated by the *LLM-Generators* in Jahan et al. (2024) across three datasets: **BC5CDR**, **DDI**, and **KD-DTI**. The **Exact Match (EM)** Accuracy (higher is better, indicated by △) and the **Root Mean Squared Error (RMSE)** (lower is better, indicated by ▽) are reported. The reported score for each *LLM-Judge* is the average of their evaluations for all *LLM-Generators* within each dataset.

LLM-Judge	BC5CDR				DDI				KD-DTI			
	Structured		Unstructured		Structured		Unstructured		Structured		Unstructured	
	EM (△)	RMSE (▽)	EM (△)	RMSE (▽)	EM (△)	RMSE (▽)	EM (△)	RMSE (▽)	EM (△)	RMSE (▽)	EM (△)	RMSE (▽)
Phi-3.5-Mini-3.8B-Instruct	53.80	0.94	35.60	2.74	32.89	1.92	32.51	2.42	36.94	2.02	36.74	2.55
Qwen-2.5-7B-Instruct	67.80	0.93	44.60	2.84	43.93	1.64	43.46	2.08	61.78	1.19	55.82	2.18
LLaMA-3.1-8B-Instruct	59.20	0.85	33.00	2.67	37.70	1.99	30.37	2.43	40.38	2.44	38.22	2.46
Deepseek-R1-Qwen-7B	57.00	1.28	31.60	3.32	41.88	2.55	35.60	2.85	47.71	1.83	46.16	2.65
Deepseek-R1-LLaMA-8B	54.40	2.25	37.40	3.64	37.70	4.14	35.60	4.62	48.40	2.68	42.36	3.31
Claude-3-Haiku	67.80	**0.60**	36.80	2.67	32.46	3.01	25.13	2.80	59.28	1.22	50.04	2.22
Gemini-1.5-flash	**70.20**	0.66	48.40	2.75	43.98	1.75	39.27	2.29	41.59	1.68	41.52	2.47
GPT-4o-mini	67.20	0.73	43.20	2.72	**53.93**	**1.45**	47.12	2.14	**72.39**	**1.08**	66.35	2.13

Table 2: Performance Comparison between ***Structured*** (our proposed) and ***Unstructured*** (baseline) output format. All the responses are generated using GPT-4-Turbo as the *LLM-Generator*.

LLM-Judge	Data		EM Accuracy
	Fine-Tuning	Evaluation	
Qwen-2.5-7B-Instruct	BC5CDR	KD-DTI	75.75 (+13.97)
Qwen-2.5-7B-Instruct	KD-DTI	BC5CDR	71.40 (+3.60)
Phi-3.5-Mini-3.8B-Instruct	BC5CDR	KD-DTI	69.54 (+32.60)
Phi-3.5-Mini-3.8B-Instruct	KD-DTI	BC5CDR	64.80 (+11.00)

Table 3: Effectiveness of Domain Adaptation via Transfer Learning. The value in brackets indicates the performance gain relative to the Zero-Shot results for *Structured* output in respective evaluation datasets in Table 2.

accuracy is quite low for LLaMA-3.1-8B while being larger than Qwen-2.5-7B. In the other two datasets (DDI and KD-DTI), Qwen-2.5-7B-Instruct achieves the best result in terms of both metrics among open-source models, even outperforming reasoning-based DeepSeek-Distilled models.

For all three datasets, we find based on paired t-test that the performance difference between the *structured* and *unstructured* approach is **statistically significant** ($p < 0.05$) for both metrics.

5.3 Effectiveness of Domain Adaptation via Transfer Learning

In our prior experiments, we demonstrate that proprietary LLMs demonstrate better performance as the judge in zero-shot scenarios. However, issues such as transparency, reproducibility, and cost highlight the need for open-source *LLM-Judge*. While early work (Kim et al., 2023, 2024) demonstrates the effectiveness of training an open-source *LLM Judge*, to our best knowledge, there is no open-source *LLM-Judge* trained on biomedical data currently available. Since there is also a lack of human-annotated judgment data that prohibits the training of *LLM-Judges* for relation extraction, we investigate a domain adaptation technique via transfer learning to address this low-resource problem.

Recent research has demonstrated that language models can effectively transfer knowledge from one dataset to another (Garg et al., 2020; Laskar et al., 2022). Inspired by the idea, in this work, we also propose transfer learning from one relation extraction dataset to the other. Based on our human-annotated labels for the *Structured* output in Section 5.2, we investigate two scenarios: (i) Fine-Tuning on the BC5CDR dataset (500 samples) and Evaluation on the KD-DTI dataset (1159 samples), and (ii) Fine-Tuning on the KD-DTI dataset (1159 samples) and Evaluation on the BC5CDR dataset (500 samples). For hyperparameter tuning, we use the DDI dataset as the validation set. We fine-tune the Qwen-2.5-7B-Instruct and Phi-3.5-3.8B-

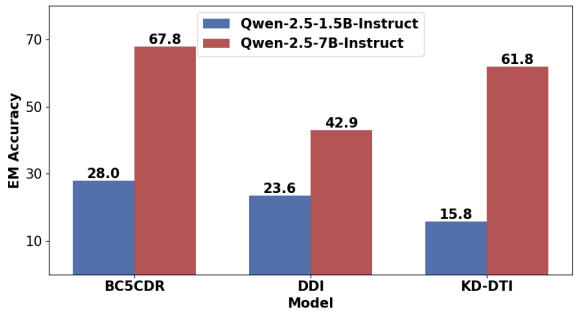

Figure 5: Performance comparison between Qwen models based on size: 1.5B and 7B, based on evaluating on the GPT-4-Turbo generated *Structured* responses.

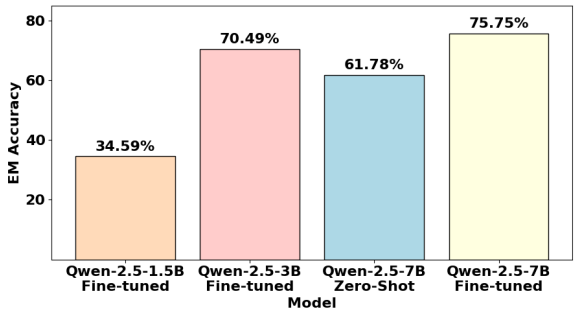

Figure 6: Comparing fine-tuned smaller models, Qwen-2.5B-Instruct (1.5B and 3B) against the larger model: Qwen-2.5B-7B-Instruct (both zero-shot and fine-tuned). The KD-DTI dataset with *Structured* response is used.

Instruct models because our previous experiments demonstrated that Qwen with 7 billion parameters outperforms the larger 8-billion parameter LLaMA model, while Phi-3.5-3.8B achieves comparable performance with the 2x larger LLaMA model.

Based on the results presented in Table 3, we find that our proposed domain adaptation strategy by transferring knowledge via fine-tuning on a limited amount of labeled data could significantly improve the accuracy. The performance gains in BC5CDR and KD-DTI by Qwen-2.5-7B-Instruct even outperform the closed-source models in Table 2. This demonstrates the effectiveness of our domain adaptation technique in low-resource scenarios.

5.4 Impact of Model Scaling

Since one of our motivations behind *LLM-Judge* is to reduce cost and improve efficiency to ensure their real-world utilization, we primarily selected open-source models having less than 10B parameters or the most cost-efficient version of different closed-source LLM providers. In this section, we investigate two scenarios:

(i) What is the impact of reducing the size of the best-performing open-source model?

(ii) Can domain adaptation via fine-tuning help even smaller models outperform larger models?

Below, we demonstrate our findings.

Reducing the Model Size: In Figure 5, we show the performance difference between the 1.5B and 7B Qwen models in zero-shot and find that reducing the model size significantly drops the accuracy. More specifically, the performance drops by 58.70%, 45.12%, and 74.44%, in BC5CDR, DDI, and KD-DTI, respectively.

Fine-tuning Smaller Models: We investigate the impact of fine-tuning smaller Qwen-2.5-Instruct models: 1.5B and 3B (we did not show their zero-shot result as they could not generate the response in the required format in most cases, leading to very poor accuracy) in comparison to the larger zero-shot model (Qwen-2.5-7B-Instruct). Based on the results shown in Figure 6, we find that the 3B model, fine-tuned on 500 samples in the BC5CDR dataset (the evaluation is conducted on KD-DTI) can outperform the zero-shot 7B model, while achieving performance comparable to the fine-tuned 7B model. This makes it possible to use smaller models in resource-constrained scenarios.

6 Conclusion and Future Work

In this paper, we evaluated the effectiveness of LLMs-as-the-Judge for assessing biomedical relation extraction models, revealing significant performance gaps between *LLM-Judges* and human evaluators. While previous work has demonstrated the potential of LLM-based evaluation for general NLP tasks, our findings indicate that existing *LLM-Judges* struggle to reliably assess biomedical relation extraction due to the nuanced and domain-specific nature of biomedical text. To improve *LLM-Judge* performance, we proposed structured output for LLM-generated responses, which led to substantial accuracy gains across multiple datasets. Additionally, we introduced a domain adaptation technique that effectively transfers knowledge from one biomedical relation extraction dataset to another to enhance the reliability of *LLM-Judges*.

In the future, we aim to explore instruction-tuning (Ouyang et al., 2022; Zhang et al., 2023) and chain-of-thought (Wei et al., 2022) prompting techniques to improve the performance of *LLM-Judges* in biomedical relation extraction. Our annotated judgment data will also be made publicly available.

Limitations

Despite the effectiveness of structured output formatting and domain adaptation, *LLM-Judges* still fall short of human evaluators. Biomedical relation extraction requires deep domain-specific reasoning, and the foundation LLMs are trained on general-domain corpora (Zhao et al., 2023; Lu et al., 2024; Minaee et al., 2024). Therefore, they still struggle with complex relationships, ambiguous terms, and implicit entity connections that human experts can recognize (some examples in Appendix A.4.2 are provided demonstrating the common errors made by *LLM-Judges* while evaluating biomedical relations extracted by different *LLM-Generators*). Nonetheless, since our human-annotated judgment dataset will be released, future work may focus on improving the performance of the LLM-Judge. Moreover, our study primarily benchmarks a limited set of LLMs, and results may vary with the release of newer or more specialized biomedical models (Singhal et al., 2023; Lu et al., 2024; Saab et al., 2024). Furthermore, cost constraints prevented the exploration of larger, more computationally expensive models, which may have improved results but are impractical in the real-world.

Ethical Considerations

Our proposed LLM-Judge is designed solely for the evaluation of LLM-generated responses in biomedical relation extraction and is not intended for direct use by end users in clinical applications. The accuracy and reliability of the relation extraction system depend on the *LLM-Generator*, which produces the relation extraction outputs. Since our proposed model only acts as an evaluator, the ethical risks associated with direct application in biomedical decision-making are minimized. By providing a scalable and efficient evaluation framework, our solution enables researchers and practitioners to quickly assess the quality of their biomedical relation extraction LLMs without relying on costly and time-consuming human evaluations. This can accelerate advancements in biomedical NLP while ensuring that models are assessed using standardized criteria. To further enhance reliability, a human-in-the-loop approach can be implemented where expert annotators verify the outputs of the models that achieve better performance based on *LLM-Judge* evaluation. Moreover, in this paper, additional compensations are not needed for the annotators since two of the authors conducted the human annotation.

References

Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, et al. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*.

Rohan Anil, Andrew M Dai, Orhan Firat, et al. 2023. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*.

Anthropic. 2023. Model card and evaluations for claude models.

Anthropic. 2024. The claude 3 model family: Opus, sonnet, haiku.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, et al. 2023. A multitask, multilingual, multimodal evaluation of ChatGPT on reasoning, hallucination, and interactivity. *arXiv preprint arXiv:2302.04023*.

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

Stella Biderman, Hailey Schoelkopf, Lintang Sutawika, Leo Gao, Jonathan Tow, Baber Abbasi, Alham Fikri Aji, Pawan Sasanka Ammanamanchi, Sidney Black, Jordan Clive, Anthony DiPofi, Julen Etxaniz, Benjamin Fattori, Jessica Zosa Forde, Charles Foster, Jeffrey Hsu, Mimansa Jaiswal, Wilson Y. Lee, Haonan Li, Charles Lovering, Niklas Muennighoff, Ellie Pavlick, Jason Phang, Aviya Skowron, Samson Tan, Xiangru Tang, Kevin A. Wang, Genta Indra Winata, François Yvon, and Andy Zou. 2024. Lessons from the trenches on reproducible evaluation of language models. *CoRR*, abs/2405.14782.

Qijie Chen, Haotong Sun, Haoyang Liu, Yinghui Jiang, Ting Ran, Xurui Jin, Xianglu Xiao, Zhimin Lin, Hongming Chen, and Zhangmin Niu. 2023. An extensive benchmark study on biomedical text generation and mining with chatgpt. *Bioinformatics*, 39(9):btad557.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Xue-Yong Fu, Md Tahmid Rahman Laskar, Elena Khasanova, Cheng Chen, and Shashi Bhushan TN. 2024. Tiny titans: Can smaller large language models punch above their weight in the real world for meeting summarization? *arXiv preprint arXiv:2402.00841*.

Siddhant Garg, Thuy Vu, and Alessandro Moschitti. 2020. Tanda: Transfer and adapt pre-trained transformer models for answer sentence selection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7780–7788.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Yuanzhuo Wang, and Jian Guo. 2024. A survey on llm-as-a-judge. *Preprint*, arXiv:2411.15594.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, et al. 2023. Textbooks are all you need. *arXiv preprint arXiv:2306.11644*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

María Herrero-Zazo, Isabel Segura-Bedmar, Paloma Martínez, and Thierry Declerck. 2013. The ddi corpus: An annotated corpus with pharmacological substances and drug–drug interactions. *Journal of biomedical informatics*, 46(5):914–920.

Yutai Hou, Yingce Xia, Lijun Wu, Shufang Xie, Yang Fan, Jinhua Zhu, Tao Qin, and Tie-Yan Liu. 2022. Discovering drug–target interaction knowledge from biomedical literature. *Bioinformatics*, 38(22):5100–5107.

Israt Jahan, Md Tahmid Rahman Laskar, Chun Peng, and Jimmy Huang. 2023. Evaluation of ChatGPT on biomedical tasks: A zero-shot comparison with fine-tuned generative transformers. In *The 22nd Workshop on Biomedical Natural Language Processing and BioNLP Shared Tasks*, pages 326–336, Toronto, Canada. Association for Computational Linguistics.

Israt Jahan, Md Tahmid Rahman Laskar, Chun Peng, and Jimmy Xiangji Huang. 2024. A comprehensive evaluation of large language models on benchmark biomedical text processing tasks. *Computers in Biology and Medicine*, page 108189.

AQ Jiang, A Sablayrolles, A Mensch, C Bamford, DS Chaplot, D de las Casas, F Bressand, G Lengyel, G Lample, L Saulnier, et al. 2023. Mistral 7b (2023). *arXiv preprint arXiv:2310.06825*.

Seungone Kim, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, et al. 2023. Prometheus: Inducing fine-grained evaluation capability in language models. In *The Twelfth International Conference on Learning Representations*.

Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2024. Prometheus 2: An open source language model specialized in evaluating other language models. *arXiv preprint arXiv:2405.01535*.

Yanis Labrak, Adrien Bazoge, Emmanuel Morin, Pierre-Antoine Gourraud, Mickael Rouvier, and Richard Dufour. 2024. Biomistral: A collection of open-source pretrained large language models for medical domains. *arXiv preprint arXiv:2402.10373*.

Md Tahmid Rahman Laskar, Sawsan Alqahtani, M Saiful Bari, et al. 2024a. A systematic survey and critical review on evaluating large language models: Challenges, limitations, and recommendations. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 13785–13816, Miami, Florida, USA. Association for Computational Linguistics.

Md Tahmid Rahman Laskar, M Saiful Bari, Mizanur Rahman, Md Amran Hossen Bhuiyan, Shafiq Joty, and Jimmy Huang. 2023. A systematic study and comprehensive evaluation of ChatGPT on benchmark datasets. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 431–469, Toronto, Canada. Association for Computational Linguistics.

Md Tahmid Rahman Laskar, Enamul Hoque, and Jimmy Xiangji Huang. 2022. Domain adaptation with pre-trained transformers for query-focused abstractive text summarization. *Computational Linguistics*, 48(2):279–320.

Md Tahmid Rahman Laskar, Elena Khasanova, Xue-Yong Fu, Cheng Chen, and Shashi Bhushan Tn. 2024b. Query-OPT: Optimizing inference of large language models via multi-query instructions in meeting summarization. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1140–1151, Miami, Florida, US. Association for Computational Linguistics.

Haitao Li, Qian Dong, Junjie Chen, Huixue Su, Yujia Zhou, Qingyao Ai, Ziyi Ye, and Yiqun Liu. 2024a. Llms-as-judges: a comprehensive survey on llm-based evaluation methods. *arXiv preprint arXiv:2412.05579*.

Jiao Li, Yueping Sun, Robin J Johnson, et al. 2016. Biocreative v cdr task corpus: a resource for chemical disease relation extraction. *Database*, 2016.

Xue Li, Fina Polat, Paul Groth, et al. 2023. Do instruction-tuned large language models help with relation extraction? In *KBC-LM/LM-KBC@ ISWC*.

Yinghao Li, Rampi Ramprasad, and Chao Zhang. 2024b. A simple but effective approach to improve structured language model output for information extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 5133–5148, Miami, Florida, USA. Association for Computational Linguistics.

Zhiyong Lu, Yifan Peng, Trevor Cohen, Marzyeh Ghassemi, Chunhua Weng, and Shubo Tian. 2024. Large language models in biomedicine and health: current research landscape and future directions. *Journal of the American Medical Informatics Association*, 31(9):1801–1811.

Renqian Luo, Liai Sun, Yingce Xia, et al. 2022. Biogpt: generative pre-trained transformer for biomedical text generation and mining. *Briefings in Bioinformatics*, 23(6).

Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. 2024. Large language models: A survey. *arXiv preprint arXiv:2402.06196*.

OpenAI. 2023. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Chengwei Qin, Aston Zhang, Zhuosheng Zhang, Jiaao Chen, Michihiro Yasunaga, and Diyi Yang. 2023. Is ChatGPT a general-purpose natural language processing task solver? *arXiv preprint arXiv:2302.06476*.

Khaled Saab, Tao Tu, Wei-Hung Weng, Ryutaro Tanno, David Stutz, Ellery Wulczyn, Fan Zhang, Tim Strother, Chunjong Park, Elahe Vedadi, et al. 2024. Capabilities of gemini models in medicine. *arXiv preprint arXiv:2404.18416*.

Karan Singhal, Shekoofeh Azizi, Tao Tu, S Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, et al. 2023. Large language models encode clinical knowledge. *Nature*, pages 1–9.

Gemini Team, Rohan Anil, Sebastian Borgeaud, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Shubo Tian, Qiao Jin, Lana Yeganova, Po-Ting Lai, Qingqing Zhu, Xiuying Chen, Yifan Yang, Qingyu Chen, Won Kim, Donald C Comeau, et al. 2024. Opportunities and challenges for chatgpt and large language models in biomedicine and health. *Briefings in Bioinformatics*, 25(1):bbad493.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45.

Congying Xia, Chen Xing, Jiangshu Du, Xinyi Yang, Yihao Feng, Ran Xu, Wenpeng Yin, and Caiming Xiong. 2024. Fofo: A benchmark to evaluate llms' format-following capability. *arXiv preprint arXiv:2402.18667*.

An Yang, Baosong Yang, Binyuan Hui, et al. 2024a. Qwen2 technical report. *Preprint*, arXiv:2407.10671.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024b. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, et al. 2024c. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, et al. 2023. Instruction tuning for large language models: A survey. *arXiv preprint arXiv:2308.10792*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623.

Zexuan Zhong and Danqi Chen. 2021. A frustratingly easy approach for entity and relation extraction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 50–61.

A Appendix

A.1 Structured Prompt for the LLM-Generator

Prompt: KD-DTI Dataset

Identify the drug-target interactions in the biomedical text given below (along with the interaction type among the following: 'inhibitor', 'agonist', 'modulator', 'activator', 'blocker', 'inducer', 'antagonist', 'cleavage', 'disruption', 'intercalation', 'inactivator', 'bind', 'binder', 'partial agonist', 'cofactor', 'substrate', 'ligand', 'chelator', 'downregulator', 'other', 'antibody', 'other/unknown'):

[BIOMEDICAL TEXT]

Note that you need to generate your response in an Array of JSON objects format with the following keys: (i) drug, (ii) target, (iii) interaction
Do not write any additional text except your response in the required JSON format.

Prompt: BC5CDR Dataset

Identify each pair of drugs (e.g., chemicals) and the drug-induced side-effects (e.g., diseases) in the following passage:

[BIOMEDICAL TEXT]

Note that you need to generate your response in an Array of JSON objects format with the following keys: (i) chemical, (ii) disease
Do not write any additional text except your response in the required JSON format.

Prompt: DDI Dataset

Identify the pairs of drug-target interactions in a given passage based on the following four interaction types:
(i) mechanism: This type is used to identify drug-drug interactions that are described by their pharmacokinetic mechanism.
(ii) effect: This type is used to identify drug-drug interactions describing an effect.
(iii) advice: This type is used when a recommendation or advice regarding a drug-drug interaction is given.
(iv) int: This type is used when a drug-drug interaction appears in the text without providing any additional information. :

[BIOMEDICAL TEXT]

Note that you need to generate your response in an Array of JSON objects format with the following keys: (i) drug, (ii) target, (iii) interaction
Do not write any additional text except your response in the required JSON format.

A.2 Models

- **GPT-4-Turbo:** It is an advanced version of OpenAI's original GPT-4 (OpenAI, 2023). The GPT-4-Turbo[9] model offers enhanced performance and efficiency, making it suitable for a wide range of applications requiring natural language understanding and generation.

- **GPT-4o-Mini:** The GPT-4o-Mini[10] is another optimized version of GPT-4. More specifically, it is a more optimized version of the recently released GPT-4o. It balances robust language understanding with efficiency. It's designed to handle complex tasks while significantly reducing operational costs.

- **Gemini-1.5-Flash:** Part of Google's Gemini-1.5 family, this model emphasizes rapid inference and the ability to handle extremely long contexts, up to one million tokens (Team et al., 2023). Gemini-1.5-Flash[11] is ideal for real-time applications where speed and processing of large amounts of data are crucial.

- **Claude-3-Haiku** Similar to GPT-4o-Mini and Gemini-1.5-Flash, it is the most cost-optimized version of the Claude-3 series (Anthropic, 2024). The Claude-3-Haiku[12] model is tailored for succinct, creative outputs. It excels at producing elegant, brief responses, while still managing complex instructions and reasoning tasks.

- **LLaMA-3.1-8B-Instruct:** This 8 billion parameter variant from Meta's LLaMA-3 (Dubey et al., 2024) series has been fine-tuned for instruction following. It strikes a balance between computational efficiency and performance, outperforming its earlier versions (Touvron et al., 2023a,b) and making it suitable for a wide range of tasks.

- **Phi-3.5-Mini-8B-Instruct:** A compact, instruction-optimized model from Microsoft's Phi series (Gunasekar et al., 2023). Despite its smaller size, it's designed to understand and execute diverse tasks in resource-constrained environments while maintaining strong performance (Abdin et al., 2024).

[9] https://platform.openai.com/docs/models/gpt-4-and-gpt-4-turbo
[10] https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/
[11] https://ai.google.dev/gemini-api/docs/models/gemini#gemini-1.5-flash
[12] https://www.anthropic.com/news/claude-3-haiku

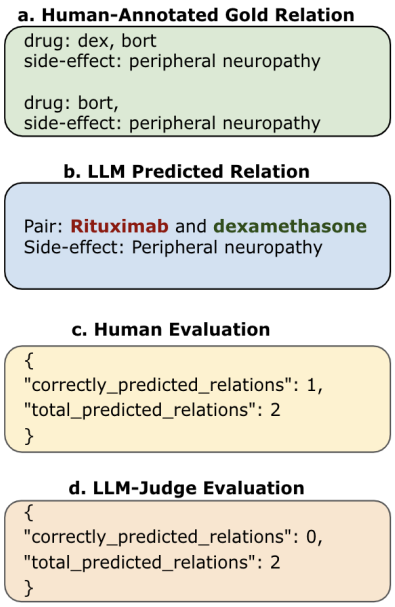

Figure 7: Our Evaluation Process where both the (c) Human Judge and the (d) LLM-Judge annotate the *correctly predicted relations* and the *total predicted relations* by comparing between (a) Human-Annotated Gold Relation and (b) LLM Predicted Relation. The biomedical text from where the relations are extracted is also provided as a context (not shown here).

- **Qwen-2.5-7B-Instruct** Developed by Alibaba, this 7-billion–parameter model is tuned for following instructions. It offers a good balance between efficiency and output quality (Yang et al., 2024a,b).

- **DeepSeek-R1-Distilled Models:** DeepSeek-R1 (Guo et al., 2025) is a reasoning model developed by DeepSeek-AI, which is mostly trained via large-scale reinforcement learning. DeepSeek-R1-Distill-Qwen-7B and DeepSeek-R1-Distill-LLaMA-8B are the distilled version of Qwen-2.5-7B-Math (Yang et al., 2024c) and LLaMA-3.1-8B-Instruct (Dubey et al., 2024), respectively, fine-tuned on 800k synthetic reasoning data generated from DeepSeek-R1.

A.3 Evaluation Process Demonstration

In Figure 7, we show our evaluation process. Since there is a mismatch between the gold human annotation and the *LLM-Judge* annotation for the *correctly predicted relations* and *total predicted relations*, the exact match score is 0. The RMSE distance is calculated as follows:

$$\sqrt{\frac{1}{2}\left((1-0)^2+(2-2)^2\right)} = \sqrt{0.5} \approx 0.71.$$

A.4 Error Analysis

A.4.1 BioMistral-7B-as-Judge Outputs

We provide some sample responses generated by BioMistral-7B model (Labrak et al., 2024), based on Mistral-7B (Jiang et al., 2023), as-the-Judge in Table 4, demonstrating its ineffectiveness as the judge by generating improper responses.

A.4.2 Error Outputs by LLM-Judges

We show some of the common error outputs of the *LLM-Judges* in Table 5. For the error output analysis, we use the overall best-performing zero-shot LLM, the GPT-4o-Mini model and demonstrate how complex biomedical terms could make it difficult for LLMs to evaluate relations extracted in an unstructured format. For instance:

- Example 1 in Table 5 demonstrates a case when 2 drugs and 6 corresponding side effects are extracted by an *LLM-Generator*, but the GPT-4o-Mini based *LLM-Judge* only considers total relations as 6 instead of 12. Moreover, it only extracted 5 as correct while the correct should be 4 side-effects for each drug (in total 8 side-effects for the 2 drugs are correct).

- Example 2 in Table 5 demonstrates a case when 2 drugs and 1 corresponding side effect for each of them have been extracted by an *LLM-Generator*. While *drug: thiopentone* and *disease (side-effects): delirium* is one of the correct answers, the *LLM-Judge* based on GPT-4o-Mini considers only 1 side effect and 1 drug. It is highly likely that GPT-4o-Mini considers *Thiopentone and propofol* as a single drug even though they should be considered different. This could be the reason it considers no correct relation is extracted. This demonstrates GPT-4o-Mini having limitations in understanding complex biomedical terms.

- Example 3 in Table 5 demonstrates a case when 2 drugs and 6 side effects in total are extracted by an *LLM-Generator*. While Pair 1 and Pair 3 have 1 side effect each, Pair 2 has 4 different side-effects. However, GPT-4o-Mini based *LLM-Judge* may have considered all these 4 side-effects as just 1 side-effect. This demonstrates its lack of biomedical text understanding capability.

Dataset	Output Format	Gold Relations	Predicted Relations	BioMistral-7B-as-Judge-Annotation
BC5CDR	Unstructured	drug: famotidine disease (side-effects): delirium	Famotidine - delirium	{ "correctly_predicted_relations": [{ "chemical": "famotidine", "disease": "delirium" }], "total_predicted_relations": 1 }
KD-DTI	Unstructured	drug: PNU156804 target: Janus kinase 3 (JAK-3) interaction: inhibitor	PNU156804 - Janus tyrosine kinase 3 (Jak3) - inhibitor PNU156804 - IL-2-mediated Jak3 auto-tyrosine phosphorylation - blocker PNU156804 - Jak3 substrates signal transducers and activators of transcription (Stat) 5a and Stat5b - blocker PNU156804 - extracellular regulated kinase 1 (Erk1) and Erk2 (p44/p42) - blocker	{ "correctly_predicted_relations": ["drug": "PNU156804", "target": "Janus kinase 3 (JAK-3)", "interaction": "inhibitor"] }
DDI	Unstructured	drug: gemfibrozil target: rosiglitazone interaction: mechanism drug: rifampin target: rosiglitazone interaction: mechanism	gemfibrozil-rosiglitazone, mechanism rifampin-rosiglitazone, mechanism CYP2C8 inhibitor/rosiglitazone, advice CYP2C8 inducer/rosiglitazone, advice	{ "correctly_predicted_relations": [1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 }
BC5CDR	Structured	drug: cyclosporine disease (side-effects): thrombotic microangiopathy drug: corticosteroids disease (side-effects): ssc drug: tacrolimus disease (side-effects):: ssc	[{ "chemical": "tacrolimus", "disease": "scleroderma renal crisis" }, { "chemical": "prednisolone", "disease": "scleroderma renal crisis" }]	{ "correctly_predicted_relations": ["chemical": "tacrolimus", "disease": "scleroderma renal crisis"], "total_predicted_relations": 2 }
KD-DTI	Structured	[{ "drug": "thiorphan", "target": "neutral endopeptidase (NEP)", "interaction": "inhibitor" }]	drug: Thiorphan target: Neutral endopeptidase (MME) interaction: inhibitor	{ "correctly_predicted_relations": [{ "drug": "Thiorphan", "target": "Neutral endopeptidase (MME)", "interaction": "inhibitor" }], "total_predicted_relations": 1, "total_gold_relations": 1, "percentage_correctly_predicted_relations": 100.0, "percentage_total_..." }
DDI	Structured	drug: CYLERT target: antiepileptic medications interaction: effect	[{ "drug": "CYLERT", "target": "other drugs", "interaction": "int" }, { "drug": "CYLERT", "target": "antiepileptic medications", "interaction": "effect" }]	{ "correctly_predicted_relations": [{ "drug": "CYLERT", "target": "antiepileptic medications", "interaction": "effect" }], "total_predicted_relations": 2 }

Table 4: Sample judgment outputs generated by BioMistral-7B in different datasets.

# Example	Gold Relations	Predicted Relations	Human Annotated Correct Relations	Human Annotated Total Relations	LLM Predicted Correct Relations	LLM Predicted Total Relations
1.	drug: s-53482 disease (side-effects): growth retardation drug: s-23121 disease (side-effects): growth retardation drug: s-23121 disease (side-effects): teratogenic drug: s-23121 disease (side-effects): embryonic death drug: s-53482 disease (side-effects): embryonic death drug: s-53482 disease (side-effects): teratogenic drug: s-23121 disease (side-effects): ventricular septal defect drug: s-53482 disease (side-effects): ventricular septal defect	Drugs: 1. S-53482 2. S-23121 Drug-induced side-effects: 1. Embryolethality 2. Teratogenicity 3. Ventricular septal defects 4. Wavy ribs 5. Growth retardation 6. Increased incidence of embryonic death It is important to note that these are drug-induced side-effects or outcomes observed in rats during the study, and further investigation is needed to understand the mechanism and its relevance to human health.	8	12	5	6
2.	drug: thiopentone disease (side-effects): postoperative delirium drug: thiopentone disease (side-effects): delirium	Pair of drugs: Thiopentone and propofol Drug-induced side-effect: Delirium	1	2	0	1
3.	drug: dox disease (side-effects): cardiotoxicity drug: dox disease (side-effects): necrosis	Pair 1: Drug - Doxorubicin Drug-induced side-effects - Cardiotoxicity Pair 2: Drug - Doxorubicin Drug-induced side-effects - Cardiac disarrangement, necrosis, DNA damage (strand breaks and oxidized pyrimidines), decreased total antioxidant performance (TAP) Pair 3: Drug - Doxorubicin Drug-induced side-effects - Increased resistance to oxidative stress	2	6	2	3

Table 5: Example of annotation errors by *GPT-4o-Mini-as-the-Judge* on some sampled data in the BC5CDR dataset.