Comparing Open-source LLMs and Human Annotators in Disease Mention Recognition

Kseniya Yerameichyk

October 13, 2024

1 Introduction

1.1 Task Description

Disease Mention Recognition (DMR) is a part of the Name Entity Recognition (NER) task that involves identifying disease names or mentions from unstructured text, such as clinical trials and medical articles. The primary goal of the DMR task is to label the spans of text that refer to diseases or medical conditions. For example, in the sentence "The patient was diagnosed with diabetes and hypertension" a DMR model would correctly identify "diabetes" and "hypertension" as disease mentions. Similarly, in a medical article like "The prevalence of Alzheimer's disease is increasing in elderly populations" the term "Alzheimer's disease" should be recognized as a disease entity.

Given the limited computational resources available for this project, we focused on the DMR task as a specific and well-defined subtask within Biomedical Named Entity Recognition. This allowed us to conduct an in-depth qualitative analysis of the LLMs performance and identify potential areas for improvement.

Considering the complexities and costs involved in manual disease annotation, we are investigating the potential of open-source LLMs to automate this process. We will adopt effective strategies described in relevant literature.

1.2 Related Works

Previous research indicates that general-domain LLMs often struggle with NER task, likely because they are primarily designed for text generation rather than sequence labeling task [6, 8]. DMR becomes an even greater challenge for the LLMs due to the complex nature of disease entities and the need for domain-specific expertise. Even state-of-the-art LLMs often underperform compared to supervised models on this task.

To address the limitations of LLMs in NER, researchers have explored transforming the task into a text generation problem. GPT-NER [1] and BioNER-LLaMA [5] have demonstrated the effectiveness of this approach, which involves generating the input text with entity tags around the found entity mentions instead of directly extracting them.

Task-specific instruction tuning has demonstrated remarkable performance in NER task [4, 5]. The BioNER-LLaMA research highlighted that general-domain LLaMA can be effectively instruction-tuned for biomedical NER, challenging the assumption that domain-specific models are always optimal. However, instruction tuning can be extremely resource-intensive. We explored prompt engineering as a potential alternative to achieve competitive results without model fine-tuning. Our findings suggest that certain prompt engineering strategies can achieve exceptional outcomes [1, 2, 8].

2 Research Question and Methodology

2.1 Project Objectives

Manual annotation of disease mentions in medical datasets is a challenging and resource-intensive process. It requires domain expertise and can be extremely time-consuming, leading to high costs of the conducted research. Moreover, the availability of qualified human annotators with the necessary medical background can be limited, complicating the creation of annotated datasets.

The goal of this project is to understand whether open-source LLMs can annotate disease datasets as accurately as human annotators with domain-specific knowledge, following the given Annotation Guidelines. By comparing the outputs of LLMs with human annotations, we can get insights about the capabilities and limitations of LLMs in this domain and explore potential applications for their use in biomedical research.

We expect that open-source LLMs will show promising results in the Disease Mention Recognition task. LLMs are likely to perform well when provided with clear annotation guidelines, given their strong ability to understand and follow instructions. However, they might struggle with more complex disease entities that require deeper medical knowledge, where human experts have an advantage.

2.2 Proposed Approach

LLMs are known for their ability to follow instructions effectively. Our approach aims to understand whether different prompt engineering techniques can successfully address the DMR task. By designing prompts thoughtfully, we want to leverage the capabilities of open-source LLMs to accurately detect disease mentions in text.

We begin with two base prompts. The markup prompt, similar to the one introduced in the Clinical NER paper [2], instructs the LLM to identify all disease mentions within the text and output the text with entities marked up using the <entity type=disease></entity> tag. This approach aligns with the idea of transforming entity recognition from a sequence labeling task into a text generation problem. However, some studies [4, 6, 7, 8] suggest that open-source LLMs can be trained to effectively extract entities from text. Therefore, the second base prompt asks the LLM to identify and extract all disease mentions and output them as a json object with the following structure "'json:"diseases": "a semicolon-separated list of specific disease names or disease classes."' Both base prompts provide one example (the same for both) since few-shot inference generally outperforms zero-shot inference [3].

Previous research [2] has shown that using dataset annotation guidelines used for manual annotation can enhance LLMs ability to accurately recognize entities. Since our project aims to evaluate if open-source LLMs can perform high-quality disease dataset annotation using the same tools as human experts, we created a prompt that includes these guidelines.

To create prompts for markup and extraction that include annotation guidelines, we adopted the guidelines from two widely used disease corpora: NCBI [13] and BC5CDR [14]. These datasets share a similar structure and follow closely aligned rules on what should and should not be annotated [15]. After reviewing all the rules, we selected eight key guidelines that we believe will most effectively improve the LLMs inference performance. The prompt annotation guidelines contain 5 positive rules (indicate the LLM what to tag or extract) and 3 negative rules (indicate what not to tag or extract):

1. Tag/Extract Multiple Disease Mentions together if cannot be separated.
2. Tag/Extract ONLY the Disease Mention when it modifies other concepts.
3. Tag/Extract All Disease Mentions, even if repeated.
4. Tag/Extract the Full, Specific Disease Mention.
5. Separate Tags for Disease Full Name and its Abbreviation/Extract Disease Full Names and Abbreviations Separately.
6. Don't tag/extract Organism Names (e.g., species, viruses, bacteria) unless they are a critical part of a Disease Name.
7. Don't tag/extract General Terms like "disease", "syndrome", "deficiency", "complications", "abnormalities", "mutation", etc. However, terms such as "cancer" and "tumor" should be tagged/extracted.
8. Don't tag/extract Biological Processes like "tumorigenesis", "cancerogenesis" etc.

The dataset annotation guidelines for human annotators include examples to clarify the rules. Research has emphasized the importance of using examples during LLMs inference for better instruction comprehension [2, 3]. The GPT-NER paper demonstrated that 3-shot inference significantly improves LLMs ability to solve NER tasks and recommended carefully selecting examples that resemble the input [1]. Therefore, we generated three examples for each rule, ensuring that they correspond to the whole NCBI and BC5CDR datasets annotation guidelines.

2.3 Used Models

We chose to use open-source LLMs for the DMR task due to their accessibility and adaptability to prompt engineering techniques. Open-source models provide visibility into their architecture, making it easier to understand how they process information and make adjustments to improve performance. By using open-source LLMs, we can leverage the power of prompt engineering to align the outputs with human expert annotation standards.

Given our limited resources, we focused on smaller LLMs (7-9 billion parameters) as they offer the best balance between size and performance. Despite being classified as "small" these models still require around 40GB of GPU RAM for inference. However, they provide a simpler inference process, making them a practical choice for high-quality results while while minimizing computational demands. After carefully evaluating existing models, we narrowed down our focus to four LLMs that had shown promising results in diverse NLP applications. A detailed description of the models is provided below.

Llama3.1 8B: Llama 3.1 8B [9] is an open-source LLM developed by Meta and released in July 2024. It has been trained on an extensive dataset of over 15 trillion tokens sourced from publicly available online platforms. This version also features an expanded context window, enabling it to handle longer inputs more effectively compared to its previous versions.

Gemma2 9B: Gemma 2 9B [10] is a compact state-of-the-art open-source LLM developed by Google and released in June 2024. It is designed as a text-to-text decoder-only model. Both the pre-trained and instruction-tuned versions have their weights publicly available. The 9B model was trained on a substantial dataset of 8 trillion tokens.

Phi3 Small 7B: Phi 3 Small 7B [11] is a decoder-only LLM developed by Microsoft and released in May 2024. It has a default context length of 8192 tokens. The training data was carefully chosen to optimize performance for smaller models by filtering publicly available web data to ensure it contained the appropriate level of "knowledge". This approach allowed the Phi 3 Small developers to retain web pages that can enhance the model reasoning abilities.

Mistral0.3 7B: Mistral0.3 7B [12] is a highly discussed decoder-only language model introduced by the French startup Mistral AI in October 2023. With 7 billion parameters, it is engineered for superior performance, outperforming leading 13B models across various benchmarks. Mistral 7B has a context length of 8192 tokens.

3 Experimental Results

3.1 Dataset

At the beginning of the experimental part we utilized a sample of 100 abstracts from the NCBI disease dataset. While we were able to perform models inference using two base prompts, we met difficulties attempting to run prompts with annotation guidelines on this dataset.

Our research prioritizes qualitative analysis rather than quantitative metrics. We aim to understand whether LLMs can comprehend and adjust to the annotation guidelines, rather than simply calculating strict or partial F1 scores based on model results. LLMs often modify the input text, even when instructed to simply add entity tags. This can lead to quantitative analysis based on entity spans overlooking important information that manual analysis can capture. This qualitative approach is crucial for assessing the models ability to follow complex instructions similar to those used by human annotators. To get this understanding we conducted extensive manual analyses to observe how modifications to the prompts (especially the annotation guidelines part) affected the models comprehension and performance.

Unfortunately, the NCBI dataset lacks abstracts containing disease entities that align with all eight rules. To ensure sufficient examples for evaluating each rule, we would have needed to perform inference on a massive number of abstracts, with no guarantee of finding enough entities to assess specific rules. Manual analysis of such a large dataset would be extremely time-consuming and impractical.

To address the described problem we decided to create an Artificial Disease Dataset that has the same structure as the NCBI dataset. This synthetic dataset includes specific entities in each abstract to ensure that all eight rules could be evaluated. The artificial dataset contains ten abstracts, which makes the manual analysis feasible. We generated the dataset by prompting ChatGPT with examples from the NCBI dataset and then manually adjusting the abstracts to guarantee that each contained entities relevant to every rule. This dataset provides us with better control for understanding the impact of the annotation guidelines on

the models results. To maintain the fairness of the experiment the disease mentions in the artificial dataset differ from those in the annotation guidelines prompt examples.

3.2 Preliminary qualitative analysis

3.2.1 Inference Mode

Before addressing our primary research question of whether LLMs can effectively follow annotation guidelines to be effective in solving DMS task, we addressed important preliminary question of whether to perform inference on entire abstracts or to split them into sentences and run inference individually.

To answer this question we conducted inference on all four LLMs using the base prompts with abstracts from the NCBI dataset. We ran two types of inference for each model: one where the input was the base prompt with the entire abstract, and another where the input was the base prompt with individual sentences (applied to all sentences of the abstract). Since we used two base prompts, this resulted in four different sets of results for each model: the markup prompt on full abstracts, the markup prompt on sentences, the extract prompt on full abstracts, and the extract prompt on sentences.

The inference results with the base prompts revealed that all models performed better when abstracts were split into sentences, with the most significant improvement observed using the extract prompt. With single-sentence inputs the models tended to generate more true positives (TP) and Partial TPs (where the entity span was not exact but had some overlap).

Since our primary focus is on the LLMs ability to correctly identify as many relevant entities as possible, we decided to conduct all further experiments on abstracts split into sentences. It allowed us to maximize the detection of true disease mentions. This approach also aligns with those used in GPT-NER [1] and BioNER-LLaMA [5], where abstracts were first split into sentences to avoid exceeding the models context window size.

3.2.2 Positive rules effect analysis

To evaluate the impact of including annotation guidelines in the prompt, we began by analyzing the effect of adding only the positive rules. We expected that including guidelines on what to tag or extract would result in more accurate entity identification, leading to an increase in TPs and Partial TPs. This initial comparison on the Artificial Disease Dataset was conducted using the base prompt and a prompt which includes the first five rules from section 2.2.

As shown in Tables 1, 2, 3 and 4, there is a significant improvement in the number of correctly found disease mentions (TPs), along with a corresponding reduction in FNs after the addition of annotation rules into the prompts for both the markup and extract prompts. This suggests that annotation guidelines can enhance the open-source LLMs ability to accurately recognize disease mentions. The markup prompt, in particular, showed the most notable improvement with an increase in TPs and a decrease in Partial TPs, indicating the models improved ability to accurately detect the full span of disease mentions. However, it is also seen that all models generate a high number of FPs, likely due to the well-known hallucination issue in LLMs, where they confidently tag or extract entities that are not actual disease mentions. We suppose that adding negative rules will help address this issue.

Llama 3.1	Partial TPs	TPs	FPs	FNs
markup (base)	29	56	46	26
markup (5 rules)	26	73	27	12
extract (base)	17	64	13	30
extract (5 rules)	22	70	14	19

Table 1: Performance metrics for **Llama 3.1** on base markup, markup with 5 rules, base extract, extract with 5 rules prompts

The key aspect of the qualitative analysis is to determine whether the models considered the annotation rules. To get a better understanding we created Tables 5, 6, 7 and 8 for each model. These tables display the number of abstracts in which the model respects the rules specified in the column headers.

Gemma 2	Partial TPs	TPs	FPs	FNs
markup (base)	16	64	16	31
markup (5 rules)	15	79	12	17
extract (base)	21	58	12	32
extract (5 rules)	19	76	4	16

Table 2: Performance metrics for **Gemma 2** on base markup, markup with 5 rules, base extract, extract with 5 rules prompts

Phi 3 Small	Partial TPs	TPs	FPs	FNs
markup (base)	50	43	21	18
markup (5 rules)	23	73	19	15
extract (base)	26	52	9	33
extract (5 rules)	19	66	15	26

Table 3: Performance metrics for **Phi 3 Small** on base markup, markup with 5 rules, base extract, extract with 5 rules prompts

Mistral 0.3	Partial TPs	TPs	FPs	FNs
markup (base)	44	53	57	14
markup (5 rules)	33	69	59	9
extract (base)	34	46	41	31
extract (5 rules)	29	58	24	24

Table 4: Performance metrics for **Mistral 0.3** on base markup, markup with 5 rules, base extract, extract with 5 rules prompts

It can be seen that rule number 5 (instructs the separation of full disease mentions from their abbreviations) is the clearest for all models, particularly with the markup prompt. Rules number 4 (tagging or extracting the full, specific disease mention) and number 2 (tagging or extracting only the disease name when it modifies another concept) appear to be well understood by the models even without explicit mention, as the annotation guidelines did not lead to any improvements in these categories of disease entities. Meanwhile, rule number 3 (tagging or extracting disease mentions even if duplicated) seems unclear to the LLMs, showing minimal impact. Only Llama 3.1 and Phi 3 Small demonstrate a slight improvement of 3 abstracts (on extract prompt) where the rule was respected, with one additional abstract improving under the markup prompt. This suggests that the rule may need to be rephrased for better clarity. The first rule was only understood by Llama 3.1 model, which showed an improvement in 6 abstracts with the markup prompt. However, the other models did not comprehend this rule, indicating a need to revise its formulation.

Llama 3.1	**1**	**2**	**3**	**4**	**5**
markup (base)	1	10	8	2	1
markup (5 rules)	7	10	9	5	10
extract (base)	2	10	0	8	0
extract (5 rules)	2	10	3	8	6

Table 5: The number of abstracts complied with annotation rules (column titles are the rules numbers) by **Llama 3.1** across different prompt types

Gemma 2	**1**	**2**	**3**	**4**	**5**
markup (base)	2	10	3	7	0
markup (5 rules)	3	10	3	7	10
extract (base)	1	10	0	7	0
extract (5 rules)	3	10	1	7	7

Table 6: The number of abstracts complied with annotation rules (column titles are the rules numbers) by **Gemma 2** across different prompt types

Phi3 Small	1	2	3	4	5
markup (base)	2	9	6	7	0
markup (5 rules)	3	8	7	6	9
extract (base)	0	10	0	8	0
extract (5 rules)	3	9	3	8	7

Table 7: The number of abstracts complied with annotation rules (column titles are the rules numbers) by **Phi 3 Small** across different prompt types

Mistral 0.3	1	2	3	4	5
markup (base)	4	8	5	8	0
markup (5 rules)	3	7	4	8	7
extract (base)	1	9	0	9	0
extract (5 rules)	1	10	0	9	3

Table 8: The number of abstracts complied with annotation rules (column titles are the rules numbers) by **Mistral 0.3** across different prompt types

3.2.3 Positive and negative rules effect analysis

To resolve the hallucination issue, we added to the prompts also negative rules 6-8 from Section 2.2, which instruct the LLMs on what not to tag or extract. We expect this will reduce the number of FPs and lead to a better understanding of the rules across a larger set of abstracts. Similar to the positive rules the negative rules section of the prompt includes three examples for each rule.

Llama 3.1	6	7	8
markup (5 rules)	10	5	3
markup (8 rules)	9	0	3
extract (5 rules)	7	5	5
extract (8 rules)	9	4	6

Table 9: The number of abstracts complied with negative annotation rules by **Llama 3.1** across different prompt types

Gemma 2	6	7	8
markup (5 rules)	6	2	8
markup (8 rules)	7	3	7
extract (5 rules)	3	8	9
extract (8 rules)	6	7	7

Table 10: The number of abstracts complied with negative annotation rules by **Gemma 2** across different prompt types

Unfortunately, the results in Tables 9, 10, 11 and 12 indicate that the LLMs do not understand the negative rules as effectively as the positive ones. For instance, in the case of the LLaMA 3.1 model with the markup prompt, the inclusion of rule 7 (do not tag general terms) actually resulted in a performance drop. Without the rule, it was followed in half of the abstracts, but with the rule, compliance dropped to zero. The only model that showed some improvement with the extract prompt after adding the negative rules was Phi 3 Small, where rule 6 (do not extract species names) and rule 8 (do not extract biological processes) saw an increase of 4 and 5 abstracts, respectively. This improvement could be explained by the fact that Phi 3 Small developers focused on the quality of data used during pre-training.

4 Concluding Remarks

4.1 Conclusion

Several key conclusions can be made from the preliminary results of this project. First, the addition of positive annotation guidelines (specifying what to tag or extract) led to a

Phi 3 Small	6	7	8
markup (5 rules)	6	1	7
markup (8 rules)	7	3	8
extract (5 rules)	4	7	5
extract (8 rules)	8	7	10

Table 11: The number of abstracts complied with negative annotation rules by **Phi 3 Small** across different prompt types

Mistral 0.3	6	7	8
markup (5 rules)	3	0	1
markup (8 rules)	5	0	1
extract (5 rules)	4	2	3
extract (8 rules)	4	3	3

Table 12: The number of abstracts complied with negative annotation rules by **Mistral 0.3** across different prompt types

significant increase in correctly identified disease mentions. This indicates that open-source LLMs can be trained to annotate disease datasets using the same tools as human annotators. Second, adding negative annotation guidelines (specifying what not to tag or extract) did not help reduce the LLMs tendency to generate a high number of FPs. This suggests that alternative approaches may be needed to address this issue.

4.2 Future Work

For future work, the first priority should be rephrasing the rules that were unclear to all four models. For the rules that were understood by only one model, additional experiments should be conducted to assess the consistency of the results, given the probabilistic nature of LLMs.

To address the hallucination issue and reduce the number of FPs we can explore a method suggested in the literature: implementing a self-verification prompt [1]. In this approach the LLM is asked to verify whether the identified entity actually belongs to the specified entity class, which is a disease name in our case.

References

[1] Wang, Shuhe, et al. "Gpt-ner: Named entity recognition via large language models." arXiv preprint arXiv:2304.10428 (2023).

[2] Hu, Yan, et al. "Improving large language models for clinical named entity recognition via prompt engineering." Journal of the American Medical Informatics Association (2024): ocad259.

[3] Monajatipoor, Masoud, et al. "LLMs in Biomedicine: A study on clinical Named Entity Recognition." arXiv preprint arXiv:2404.07376 (2024).

[4] Zhou, Wenxuan, et al. "Universalner: Targeted distillation from large language models for open named entity recognition." arXiv preprint arXiv:2308.03279 (2023).

[5] Keloth, Vipina K., et al. "Advancing entity recognition in biomedicine via instruction tuning of large language models." Bioinformatics 40.4 (2024): btae163.

[6] Biana, Junyi, et al. "VANER: Leveraging Large Language Model for Versatile and Adaptive Biomedical Named Entity Recognition." arXiv preprint arXiv:2404.17835 (2024).

[7] Luo, Ling, et al. "AIONER: all-in-one scheme-based biomedical named entity recognition using deep learning." Bioinformatics 39.5 (2023): btad310.

[8] Ashok, D., and Z. C. Lipton. "PromptNER: Prompting For Named Entity Recognition. arXiv 2023." arXiv preprint arXiv:2305.15444.

[9] Dubey, Abhimanyu, et al. "The llama 3 herd of models." arXiv preprint arXiv:2407.21783 (2024).

[10] Team, Gemma, et al. "Gemma 2: Improving open language models at a practical size." arXiv preprint arXiv:2408.00118 (2024).

[11] Abdin, Marah, et al. "Phi-3 technical report: A highly capable language model locally on your phone." arXiv preprint arXiv:2404.14219 (2024).

[12] Jiang, Albert Q., et al. "Mistral 7B." arXiv preprint arXiv:2310.06825 (2023).

[13] Doğan, Rezarta Islamaj, Robert Leaman, and Zhiyong Lu. "NCBI disease corpus: a resource for disease name recognition and concept normalization." Journal of biomedical informatics 47 (2014): 1-10.

[14] Li, Jiao, et al. "BioCreative V CDR task corpus: a resource for chemical disease relation extraction." Database 2016 (2016).

[15] Dogan, Rezarta Islamaj, and Zhiyong Lu. "An improved corpus of disease mentions in PubMed citations." BioNLP: Proceedings of the 2012 Workshop on Biomedical Natural Language Processing. 2012.