SCIRE at BioASQ 2025: LLM Driven Biomedical Named Entity Recognition for GutBrainIE 2025

Harsh Prakash Gupta[1,*], Ritwik Banerjee[1,*]

[1]Stony Brook University, Stony Brook, New York, USA

Abstract

In recent years, we have witnessed the rise of powerful Large Language Models (LLMs) and their flexibility in accomplishing a wide range of NLP tasks, often achieving state-of-the-art (SOTA) accuracy. However for Named Entity Recognition (NER), there is a specific need for token-level class assignments and bidirectional context, both preceding and following a token, to understand its role. As a result, bidirectional encoder-style transformer models (BERT-like models) have been the standard approach. However, fine-tuning these models on available datasets often faces the bottleneck of limited training data. In this paper, we propose an alternative approach that leverages the extensive knowledge base of decoder-style Transformer models. These modern LLMs are typically trained on vast amounts of text, which enables them to overcome the challenge of limited labeled data. Instead of training from scratch, we focus on aligning the responses of these LLMs to suit NER. To this end, we use two methods: (i) few-shot prompting, and (ii) fine-tuning on available examples. Our findings indicate that fine-tuning significantly outperforms prompting for biomedical NER, effectively aligning LLM outputs to the desired task outputs. Additionally, we propose an algorithm to parse the output of the LLM to extract relevant entities, their labels, and their start and end indices.

Keywords

Large language models, Named entity recognition, Transformer, BERT, GPT, OpenAI

1. Introduction

The GutBrainIE2025 challenge [1] presented a set of four tasks focused on extracting structured information from biomedical texts. The overarching theme of the challenge centers on identifying relationships between the gut microbiota and conditions such as Parkinson's disease and mental health disorders. GutBrainIE2025 is the sixth task of the BioASQ [2] CLEF Lab 2025. In this paper, we focus exclusively on the first task of the challenge (Subtask 6.1): *named entity recognition* (NER). For this subtask, participants were provided with a collection of PubMed [3] articles, including their titles and abstracts. The objective was defined as the identification and classification of entities into one of the following predefined categories: *Anatomical Location*, *Animal*, *Biomedical Technique*, *Bacteria*, *Chemical*, *Dietary Supplement*, *Disease*, *Disorder or Finding (DDF)*, *Drug*, *Food*, *Gene*, *Human*, *Microbiome*, and *Statistical Technique*. For each entity, it was also required that the entity's text span must be identified — that is, its start and end character indices, and whether it appears in the title or the abstract.

A competitive baseline model was provided to participants. This was a fine-tuned version of the NuNER Zero model [4] developed by NuMind. The model is a compact encoder-based Transformer designed for zero-shot NER. It is based on the GLiNER [5] architecture, enabling it to identify entities without requiring task-specific fine-tuning. Given that the baseline already uses a capable encoder-style Transformer model, we explore the alternative approach, where we use decoder-based Transformer models. Specifically, we employ Large Language Models (LLMs). These models have demonstrated significant advancements and are typically trained on massive corpora, including biomedical texts. This makes them promising candidates for tasks like biomedical NER. In this work, we investigate whether LLMs can match or surpass the performance of encoder-based models in the biomedical NER task.

CLEF 2025 Working Notes, 9 – 12 September 2025, Madrid, Spain

*Corresponding author.

hpgupta@cs.stonybrook.edu (H. P. Gupta); rbanerjee@cs.stonybrook.edu (R. Banerjee)

0009-0003-3415-2715 (H. P. Gupta); 0000-0003-0336-0258 (R. Banerjee)

2. Related Work

Recent advances in large language models (LLMs) have demonstrated significant promise for named entity recognition (NER) tasks across diverse domains. Several studies have explored the application of LLMs to NER, with notable successes in both general and specialized contexts. Xie et al. [6] empirically investigated the use of ChatGPT [7] for zero-shot named entity recognition, demonstrating improvements across seven benchmarks, while the LTNER framework [8] employed contextualized entity marking to enhance LLM performance on NER tasks. Their primary results (91.9% F1) were, however, reported on the older benchmark of CoNLL03 [9]. In the biomedical domain, comparative studies have revealed that certain LLMs, including open-source LLMs like Mistral [10] and Llama [11], can outperform traditional encoder models like BERT on biomedical NER, particularly for longer entities [12]. This improvement, however, was marginal.

Chen et al. [13] provided a comprehensive systematic evaluation of four large language models (GPT-3.5, GPT-4, LLaMA 2, and PMC LLaMA) across 12 biomedical NLP datasets, including named entity recognition tasks under various settings. Additionally, domain-specific applications have shown promising results, with Jung et al. [14] demonstrating high performance using LLM-based biological named entity recognition on scientific literature related to p53 protein research, and Bian et al. [15] achieving strong one-shot biomedical named entity recognition performance through a two-step approach incorporating domain-specific knowledge.

2.1. Adapting NER for Large Language Models

When using Large Language Models (LLMs) for the NER task, it is crucial to adapt the task format in a way that aligns with the natural strengths of LLMs. Since LLMs are fundamentally next-token predictors, they are not inherently well-suited for pinpointing the exact start and end indices of entity spans, even if they can correctly identify and classify the entities themselves. Therefore, it becomes necessary to reformulate NER as a generative task, a paradigm where LLMs excel.

GPT-NER [16] introduced a novel annotation strategy that adapts NER to suit LLMs. It transforms the task into a token prediction problem by marking the beginning and end of an entity span with special tokens: `@@` for the start and `##` for the end. The entity's class label immediately follows the span-end marker `##`, resulting in the format: `@@<text_span>##<class_label>`. This structured output makes it straightforward to parse the model's predictions and extract entities along with their labels. While GPT-NER focuses solely on entity recognition and classification, without requiring start and end character indices, the GutBrainIE challenge [1] does require those indices. Consequently, we developed a custom parsing technique to accurately extract entity boundaries along with their associated metadata. Additionally, GPT-NER did not involve fine-tuning the LLM. Instead, it employed few-shot prompting along with a self-verification mechanism, wherein the LLM was prompted to assess whether the extracted spans matched any known entity class. This approach was tested using GPT-3, which in 2023 did not support fine-tuning for external users. Despite this, GPT-NER achieved performance levels comparable to fully supervised baselines.

3. Dataset

The organizers provided four datasets with varying annotation quality: Platinum, Gold, Silver, and Bronze, in descending order of reliability. The Bronze annotations were automatically generated using the fine-tuned baseline model. Since the Platinum and Gold datasets together included a sufficient number of annotated articles (111 + 208 = 319), we used their combination as our training set. We excluded the Silver and Bronze datasets to maintain high data quality.

4. Methodology

There are currently many strong LLMs available, both open-source and proprietary, accessible via APIs. For this challenge, we chose the GPT-4.1-mini from OpenAI, for several reasons: the GPT-4.1 series excels at instruction following, code generation, and handling long-context tasks; and importantly, GPT-4.1-mini is also significantly more cost-effective at approximately 84% cheaper than GPT-4o, while achieving nearly equivalent scores on benchmarks such as MMLU [17].

4.1. Annotation Format and Prompting Strategy

We adopted the annotation strategy introduced by GPT-NER [16]. Each input to the LLM consisted of a text segment (either from the title or abstract of a PubMed article) along with the list of 13 entity classes. The LLM was instructed to reproduce the original input text exactly, with the only modification being the insertion of entity annotations in the following format: @@<text_span>##<class_label>. The instructions emphasized that all characters—letters, spaces, and special Unicode characters—must be preserved exactly, apart from the added annotation tokens. This consistency is critical for accurately calculating the start and end character indices of each entity.

4.2. Few-Shot Prompting

To provide the model with relevant in-context examples, we moved beyond a fixed set of few-shot examples and instead implemented a dynamic Retrieval-Augmented Generation (RAG) strategy. This approach ensures that the examples included in the prompt are contextually similar to the input text, thereby offering more targeted guidance to the LLM. The RAG process involved three main steps:

4.2.1. Indexing of Training Examples

First, we processed the entire training set (319 articles) to create a searchable knowledge base. For each title and abstract in the training data, we generated a dense vector embedding using OpenAI's `text-embedding-3-large` model. These embeddings, along with their corresponding text and metadata (PubMed ID, example ID), were then indexed using the FAISS (Facebook AI Similarity Search) [18] library, an efficient similarity search library developed by Meta. We specifically utilized an `IndexFlatL2` index, which performs an exhaustive search by calculating the L2 (Euclidean) distance between the query vector and all vectors in the index. While computationally intensive for massive datasets, this index guarantees the retrieval of the exact nearest neighbors, making it ideal for our moderately-sized corpus where accuracy is paramount.

4.2.2. Real-time Retrieval of Relevant Examples

When a new input text (from the validation or test set) required annotation, we first generated its embedding using the same `text-embedding-3-large` model. This query embedding was then used to search the FAISS index to retrieve the top-k most similar examples from our training set.

4.2.3. Dynamic Prompt Construction

The retrieved examples were then used to construct the final prompt for the LLM. For each retrieved example, we generated the expected annotated output based on the ground-truth labels. These input-output pairs were then prepended to the prompt that contained the new text to be annotated. The number of examples included was based on the prompting configuration:

- **0-shot:** No examples were retrieved or included.
- **1-shot:** The single most similar abstract example was included.

- **3-shot:** The top two most similar abstract examples and the single most similar title example were included.
- **5-shot:** The top three most similar abstract examples and the top two most similar title examples were included.

This balance ensured the model was exposed to both abstract and title texts, which differ in length and structure along with this RAG-based strategy ensured that the model was always primed with highly relevant examples that mirrored the structure and content of the target text. However, as detailed in the Results section, this dynamic few-shot approach still yielded significantly worse performance than the baseline model, which led us to explore fine-tuning.

4.3. Fine-Tuning

As `gpt-4.1-mini` is a proprietary model, we don't have access to its weights that we can manipulate ourselves using standard finetuning process, thus we have to use OpenAI's fine-tuning framework, which requires a .jsonl file containing input-output pairs. For each training example, we used the entity spans and labels to generate the expected annotated output in the same format used during prompting. Both training and validation sets were prepared and uploaded to OpenAI's API. Once fine-tuning was complete, the resulting model could be accessed via a dedicated API endpoint, just like OpenAI's standard models. Post fine-tuning, we re-evaluated the model using the same prompting configurations (0, 1, 3, and 5-shot) to measure the improvements in entity recognition accuracy.

OpenAI's fine-tuning framework allows users to configure a limited set of hyperparameters. The three adjustable hyperparameters are: Batch size, Number of epochs, Learning rate multiplier. These can either be manually specified or set to "auto", allowing OpenAI to select optimal values automatically. Through experimentation, we identified a configuration that produced the best results for our use case: The training prompt for each example included only the base instruction (i.e., 0-shot setup) and excluded any in-context examples. Including 1-shot, 3-shot, or 5-shot examples during fine-tuning negatively impacted recall and also increased token usage, leading to higher training costs. The train set included examples only from Gold and Platinum sets from the dataset provided by GutBrainIE2025 [1] and thus we had 319 in total examples where each example contained an abstract part and a title part. Both batch size and learning rate multiplier were set to "auto", which OpenAI internally configured as 1 for our training job. We observed that increasing the number of epochs reduced training loss but increased validation loss, which indicates potential overfitting. The best validation performance was achieved at the end of epoch 1, and all results reported in the next section (Sec. 5) are from this checkpoint.

4.4. Parsing LLM Output

The output generated by the LLM follows the annotation format described earlier. For each identified entity, the challenge requires extracting five key pieces of information: Location (title or abstract), Class label, Text span (the exact string constituting the entity), Start index (in the original text), End index (in the original text).

To achieve this, we developed a parsing algorithm that converts the annotated text into structured entity objects with these five attributes. Although the LLM is explicitly instructed to only insert the special tokens @@, ##, and the corresponding class label—while keeping all other characters unchanged—hallucinations can still occur, particularly with non-fine-tuned models. We observed such deviations only in outputs from the base GPT-4.1-mini, not in the fine-tuned version. Nevertheless, our parsing approach is designed to be robust against such inconsistencies and to recover valid entities wherever possible.

4.4.1. Step 1: Preprocessing

We first remove any malformed or extraneous occurrences of `@@` or `##` that do not conform to the valid pattern: `@@<text_span>##<class_label>`. This is done using regular expressions.

Table 1
Performance on Validation Set

Model	Macro-P	Macro-R	Macro-F1	Micro-P	Micro-R	Micro-F1
Baseline	0.6627	0.7473	0.6917	0.7561	**0.8272**	0.7901
Gpt-4.1-mini 0 shot	0.3527	0.3957	0.3549	0.4172	0.4602	0.4376
Gpt-4.1-mini 1 shot	0.3974	0.4659	0.4168	0.4951	0.5470	0.5198
Gpt-4.1-mini 3 shot	0.4145	0.4950	0.4384	0.5140	0.5900	0.5494
Gpt-4.1-mini 5 shot	0.4367	0.5529	0.4801	0.5292	0.6168	0.5697
Gpt 4.1-mini finetune 0 shot	0.7855	**0.7474**	**0.7486**	0.8324	0.8227	**0.8276**
Gpt 4.1-mini finetune 1 shot	**0.8006**	0.7160	0.7249	**0.8376**	0.8174	0.8274
Gpt 4.1-mini finetune 3 shot	0.7524	0.7028	0.6995	0.8203	0.8093	0.8148
Gpt 4.1-mini finetune 5 shot	0.7727	0.6952	0.7044	0.8332	0.8048	0.8188

4.4.2. Step 2: Pattern Matching

W again use regular expressions to extract all valid entity annotations from the output text. From each match, we extract: Group 1: the entity's text span, and Group 2: the associated class label. To calculate the character indices, we maintain a cumulative offset variable that tracks the total number of extra characters (@@, ##, and class labels) that have been inserted into the original text. Subtracting this offset from the indices in the annotated text, we can extract the text span.

4.4.3. Step 3: Index Correction

In cases where hallucinations occur, the calculated indices might not correctly align with the original text—typically because the entity string has been slightly altered. To recover from this, we apply a sliding window search strategy: First, we attempt to match the extracted text span in the original text by sliding it rightward up to 10 characters. If no match is found, we slide it leftward up to 10 characters. If a match is found during this process, we use that position as the corrected start index. If no match is found at all, we adjust the offset to account for the mismatch and proceed to the next entity. This combination of pattern matching, offset correction, and sliding window search ensures that we can reliably extract entity information even in the presence of minor hallucinations or deviations from the expected output format.

4.5. Token Usage and Inference Cost

Our training set consisted of 319 articles, each containing both a title and an abstract. This yielded a total of 638 training examples (319 × 2). The total token count—including both the base prompt and annotated output—was approximately 500,000 tokens. OpenAI's pricing for fine-tuning the GPT-4.1-mini model is approximately $5 per 1 million tokens. Therefore, training for one epoch (500K tokens) cost around $2.50. Inference with both the base and fine-tuned versions of GPT-4.1-mini is highly cost-effective, averaging about $1 per 1 million tokens, making this approach scalable for real-world applications.

5. Results and Analysis

All reported results in Table 1 are based on the validation set. Table 2 below shows the official evaluation results on the test set for our best-performing model: the fine-tuned GPT-4.1-mini (0-shot), as provided by the organizers. Our findings demonstrate a clear performance gap between the base and fine-tuned versions of GPT-4.1-mini.[1]

[1]All code for this work is available on GitHub at github.com/hpgupt/GutBrainIE-CLEF25.

Table 2
Official Evaluation Results on Test Set (provided by organizers)

Method	Macro-P	Macro-R	Macro-F1	Micro-P	Micro-R	Micro-F1
Gpt 4.1-mini finetune 0 shot	**0.7368**	0.7682	**0.7471**	**0.7956**	**0.8278**	**0.8114**
Baseline	0.6883	**0.7690**	0.7047	0.7639	0.8238	0.7927

5.1. Base GPT-4.1-mini (Few-shot Prompting)

The base model's performance improves progressively with the number of few-shot examples: 5-shot prompting achieved the best results, followed by 3-shot, 1-shot, and 0-shot, which performed the worst. This trend suggests that the base model, having no task-specific tuning, benefits from additional in-context examples that help it better understand the task format.

5.2. Fine-tuned GPT-4.1-mini

Interestingly, the fine-tuned model shows the reverse trend: 0-shot prompting consistently outperforms 1-shot and 5-shot methods. 5-shot prompting, in particular, performs the worst among all configurations post-fine-tuning. This reversal likely occurs because the fine-tuning process itself exposes the model to hundreds of training examples. As a result, additional in-context examples during inference do not improve performance—and may even degrade it due to token budget constraints or overfitting to redundant patterns. Consequently, the 0-shot fine-tuned GPT-4.1-mini emerges as the best-performing model overall. It also offers significant inference cost savings, since fewer tokens are consumed compared to multi-shot prompting strategies.

5.3. Understanding the Gap Between Base and Fine-Tuned Models

Even though the base model performs worse across the board, qualitative analysis reveals that its errors are often not due to a lack of understanding, but rather ambiguities in entity boundary definitions. For example, consider the training annotation: The span "secretory IgA (SIgA)" is labeled in the dataset as two separate entities: "secretory IgA" and "SIgA", both tagged as Gene. However, the base model often predicts the entire phrase "secretory IgA (SIgA)" as a single entity. While this is not incorrect from a semantic perspective, it fails to match the exact annotated boundaries, leading to penalization in standard evaluation metrics like precision and recall. The fine-tuning process helps the model learn these annotation-specific conventions, allowing it to mimic the labeling patterns found in the training set more closely. This alignment with annotator intent significantly boosts its performance metrics.

Finally, we can conclusively state that our best model—the fine-tuned GPT-4.1-mini (0-shot)—outperforms the baseline provided by the challenge organizers. However, it is important to note that the improvement in evaluation metrics is marginal. This raises an important consideration regarding efficiency vs. performance. The baseline model (a fine-tuned NuNER/GLiNER encoder) consists of millions of parameters, while GPT-4.1-mini is a much larger decoder-style model with billions of parameters. As a result, the computational cost for inference with GPT-4.1-mini is significantly higher. In summary, if maximum accuracy is the primary goal, then LLMs like GPT-4.1-mini – especially when fine-tuned – offer a promising path forward for biomedical NER. However, if efficiency and resource constraints are critical, then state-of-the-art encoder-based models like GLiNER remain highly competitive and more practical for deployment at scale.

6. Future Work

While fine-tuned LLMs such as GPT-4.1-mini demonstrate strong potential for biomedical NER, several directions remain open for future research and optimization. These include exploring smaller, open-source LLMs by applying the same fine-tuning and prompting techniques used with larger models; this

would assess whether comparable performance can be achieved with significantly lower computational cost, reduce reliance on proprietary systems, and clarify if data quality rather than model size is the primary bottleneck.

Furthermore, developing improved and more LLM-friendly annotation strategies beyond the GPT-NER style is critical, as the design of annotation schemes profoundly impacts model performance for tasks like entity recognition and span extraction. Alternatively, investigating how LLMs perform under the traditional BIO (Beginning, Inside, Outside) tagging scheme could enable structured token-level classification and potentially better align LLM outputs with existing NER pipelines.

Acknowledgments

This work was supported in part by a seed award from the AI Innovation Institute (AI3) at Stony Brook University (State University of New York at Stony Brook).

Declaration on Generative AI

During the preparation of this work, the authors used GPT-4o for grammar and spelling checks. After using this tool, the authors reviewed and edited the content as needed and takes full responsibility for the publication's content.

References

[1] M. Martinelli, G. Silvello, V. Bonato, G. M. Di Nunzio, N. Ferro, O. Irrera, S. Marchesin, L. Menotti, F. Vezzani, Overview of GutBrainIE@CLEF 2025: Gut-Brain Interplay Information Extraction, in: G. Faggioli, N. Ferro, P. Rosso, D. Spina (Eds.), CLEF 2025 Working Notes, 2025.

[2] A. Nentidis, G. Katsimpras, A. Krithara, M. Krallinger, M. Rodríguez-Ortega, E. Rodriguez-López, N. Loukachevitch, A. Sakhovskiy, E. Tutubalina, D. Dimitriadis, G. Tsoumakas, G. Giannakoulas, A. Bekiaridou, A. Samaras, G. M. Di Nunzio, N. Ferro, S. Marchesin, M. Martinelli, G. Silvello, G. Paliouras, Overview of BioASQ 2025: The thirteenth BioASQ challenge on large-scale biomedical semantic indexing and question answering, volume TBA of *Lecture Notes in Computer Science*, Springer, 2025, p. TBA.

[3] J. White, Pubmed 2.0, Medical Reference Services Quarterly 39 (2020) 382–387. doi:10.1080/02763869.2020.1826228.

[4] S. Bogdanov, A. Constantin, T. Bernard, B. Crabbé, E. Bernard, Nuner: Entity recognition encoder pre-training via llm-annotated data, arXiv preprint arXiv:2402.15343 (2024). arXiv:2402.15343.

[5] U. Zaratiana, N. Tomeh, P. Holat, T. Charnois, Gliner: Generalist model for named entity recognition using bidirectional transformer, arXiv preprint arXiv:2311.08526 (2023). arXiv:2311.08526.

[6] T. Xie, Q. Li, J. Zhang, Y. Zhang, Z. Liu, H. Wang, Empirical study of zero-shot NER with ChatGPT, in: H. Bouamor, J. Pino, K. Bali (Eds.), Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Singapore, 2023, pp. 7935–7956. doi:10.18653/v1/2023.emnlp-main.493.

[7] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, R. Lowe, Training language models to follow instructions with human feedback, 2022. doi:10.48550/arXiv.2203.02155. arXiv:2203.02155.

[8] F. Yan, P. Yu, X. Chen, LTNER: Large Language Model Tagging for Named Entity Recognition with Contextualized Entity Marking, in: Pattern Recognition: 27th International Conference, ICPR 2024, Kolkata, India, December 1–5, 2024, Proceedings, Part XIX, Springer-Verlag, Berlin, Heidelberg, 2024, p. 399–411. doi:10.1007/978-3-031-78495-8_25.

[9] E. F. Tjong Kim Sang, F. De Meulder, Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition, in: Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003, 2003, pp. 142–147.

[10] S. Karamcheti, L. Orr, J. Bolton, T. Zhang, K. Goel, A. Narayan, R. Bommasani, D. Narayanan, T. Hashimoto, D. Jurafsky, C. D. Manning, C. Potts, C. Ré, P. Liang, Mistral - A Journey towards Reproducible Language Model Training, 2021. URL: https://github.com/stanford-crfm/mistral.

[11] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al., Llama: Open and efficient foundation language models, arXiv preprint arXiv:2302.13971 (2023).

[12] M. S. Obeidat, M. S. Al Nahian, R. Kavuluru, Do llms surpass encoders for biomedical ner?, arXiv preprint arXiv:2504.00664 (2025). arXiv:2504.00664.

[13] Q. Chen, Y. Hu, X. Peng, Q. Xie, Q. Jin, A. Gilson, M. B. Singer, X. Ai, P.-T. Lai, Z. Wang, et al., Benchmarking large language models for biomedical natural language processing applications and recommendations, Nature communications 16 (2025) 3280.

[14] S. J. Jung, H. Kim, K. S. Jang, Llm based biological named entity recognition from scientific literature, in: 2024 IEEE International Conference on Big Data and Smart Computing (BigComp), 2024, pp. 433–435. doi:10.1109/BigComp60711.2024.00095.

[15] J. Bian, J. Zheng, Y. Zhang, H. Zhou, S. Zhu, One-shot Biomedical Named Entity Recognition via Knowledge-Inspired Large Language Model, in: Proceedings of the 15th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics, BCB '24, Association for Computing Machinery, New York, NY, USA, 2024. doi:10.1145/3698587.3701356.

[16] S. Wang, X. Sun, X. Li, R. Ouyang, F. Wu, T. Zhang, J. Li, G. Wang, Gpt-ner: Named entity recognition via large language models, arXiv preprint arXiv:2304.10428 (2023). arXiv:2304.10428.

[17] OpenAI, Gpt-4 turbo and the future of ai assistants, 2024. URL: https://openai.com/index/gpt-4-1/, accessed: 2025-05-19.

[18] Facebook AI Research, FAISS: Facebook AI Similarity Search, 2025. URL: https://github.com/facebookresearch/faiss, accessed: 2025-05-19.

Appendix

A. Prompts and 1-Shot Example

This section provides the detailed prompt used for instructing the LLM and a concrete example of a 1-shot prompt constructed using our Retrieval-Augmented Generation (RAG) approach. We are using a "title" example as input here as it is much shorter than "abstract".

```
Perform Named Entity Recognition on the biomedical text provided in the "Text"
section using a specific inline annotation style.

Task Description:
You are an expert Named Entity Recognition (NER) system specializing in biomedical
texts related to the gut-brain axis. Your task is to identify and extract entities
from the provided text. Identify all mentions of entities belonging to the
predefined categories listed below. An entity can occur multiple times; treat each
occurrence as a separate
entity and mark them directly within the text. Mark the beginning of an entity's
span with "@@" and the end of the span with "##" followed immediately by the
entity's category label. Note - the output text should be exactly identical to the
input text i.e all spaces , special characters/ unicode characters,html/markdown
tags,etc (any character) also
should be exactly the same, except for the added "@@, ## and entity label"
annotations for each entity detected.
Text span of an entity means the actual words/characters that form the entity in the
text. An entity's span can contain single or multiple words but never partial words.
The format to follow while marking an entity with its label is
@@<entity_text_span>##<label>

Predefined Entity Categories (use these exact labels after ##):
[
    "anatomical_location", "animal", "biomedical_technique", "bacteria", "chemical",
    "dietary_supplement", "DDF", "drug", "food", "gene", "human", "microbiome",
    "statistical_technique"
]

Note - DDF stands for Disease, Disorder, or Finding. The remaining categories refer
to their conventional or scientific meaning.
Also, If the first word or first set of words in output belong to an entity then
ensure to start the output with @@ and follow rest of instructions.

Follow the format shown in the detailed examples below precisely.
### Examples
Input: The brain-gut axis has gained increasing attention due to its contribution
to the etiology of various central nervous system disorders. This study aims to
elucidate the hypothesis that schizophrenia is associated with
disturbances in intestinal microflora and imbalance in intestinal metabolites. By
exploring the intricate relationship between the gut and the brain, with the
goal of offering fresh perspectives and valuable insights into the potential
contribution of intestinal microbial and metabolites dysbiosis to the etiology of
schizophrenia. In this study, we used a 16S ribosomal RNA (16S rRNA) gene
sequence-based approach and an untargeted liquid chromatography-mass
spectrometry-based metabolic profiling approach to measure the gut microbiome and
microbial metabolites from 44 healthy controls, 41 acute patients, and 39 remission
patients, to evaluate whether microbial dysbiosis and microbial metabolite
biomarkers were linked with the severity of schizophrenic symptoms.
```

```
Here, we identified 20 dominant disturbances in the gut microbial composition of
patients compared with healthy controls, with 3 orders, 4 families, 9 genera, and 4
species. Several unique bacterial taxa associated with
schizophrenia severity. Compared with healthy controls, 145 unusual microflora
metabolites were detected in the acute and remission groups, which were mainly
involved in environmental information processing,metabolism, organismalsystems, and
human diseases in the Kyoto encyclopedia of genes and genomes pathway. The Sankey
diagram showed that 4 abnormal intestinal and 4 anomalous
intestinal microbial metabolites were associated with psychiatric clinical symptoms.
These findings suggest a possible interactive influence of the gut microbiota and
their metabolites on the pathophysiology of schizophrenia.

Output: The brain-gut axis has gained increasing attention due to its contribution
to the etiology of various @@central nervous system disorders##DDF. This study aims
to elucidate the hypothesis that @@schizophrenia##DDF is associated with
disturbances in @@intestinal microflora##microbiome and imbalance in @@intestinal
metabolites##chemical. By
exploring the intricate relationship between the @@gut##anatomical_location and the
@@brain##anatomical_location, with the goal of offering fresh perspectives and
valuable insights into the potential contribution of @@intestinal microbial
and metabolites dysbiosis##DDF to the etiology of @@schizophrenia##DDF. In this
study, we used a @@16S ribosomal RNA (16S rRNA) gene sequence-based
approach##biomedical_technique and an @@untargeted liquid chromatography-mass
spectrometry-based metabolic profiling approach##biomedical_technique to measure
the @@gut microbiome##microbiome and
@@microbial metabolites##chemical from 44 @@healthy controls##human, 41 @@acute
patients##human, and 39 @@remission patients##human, to evaluate
whether @@microbial dysbiosis##DDF and microbial metabolite biomarkers were linked
with the severity of schizophrenic symptoms. Here, we identified 20 dominant
disturbances in the @@gut microbial composition of patients##human
compared with @@healthy controls##human, with 3 orders, 4 families, 9 genera, and 4
species. Several unique bacterial taxa associated with @@schizophrenia##DDF
severity. Compared with @@healthy controls##human, 145 unusual @@microflora
metabolites##chemical were detected in the acute and
remission groups, which were mainly involved in environmental information
processing, metabolism, organismal systems, and human diseases in the Kyoto
encyclopedia of genes and genomes pathway. The Sankey diagram showed that 4
abnormal intestinal and 4 anomalous intestinal @@microbial metabolites##chemical
were associated with @@psychiatric clinical symptoms##DDF. These findings suggest a
possible interactive influence of the
@@gut microbiota##microbiome and their metabolites on the pathophysiology of
@@schizophrenia##DDF.

### Text
Input: Alteration of Gut Microbiome in Patients With Schizophrenia Indicates Links
Between Bacterial Tyrosine Biosynthesis and Cognitive Dysfunction.
Output:
```

B. Training Instance Example

The fine-tuning dataset was structured as a ‘.jsonl‘ file, where each line is a JSON object representing a single training example. Each object contains a list of messages defining the conversation flow: a system message with basic instructions, a user message with the input text and task instructions, and an assistant message with the correctly annotated output. Below is one example from our training set, using a title for brevity.

{"messages": [{"role": "system", "content": "You are an expert Named Entity Recognition (NER) system specializing in biomedical texts related to the gut-brain axis."}, {"role": "user", "content": "\nTask Description:\nYour task is to identify and extract entities from the provided text. Perform Named Entity Recognition on the biomedical text provided in the \"Text\" section using a specific inline annotation style.\nIdentify all mentions of entities belonging to the predefined categories listed below. An entity can occur multiple times; treat each occurrence as a separate entity and mark them directly within the text.\nMark the beginning of an entity's span with \"@@\" and the end of the span with \"##\" followed immediately by the entity's category label. Note - the output text should be exactly identical to the input text i.e all spaces , special characters/ unicode characters,html/markdown tags,etc (or any other character) also should be exactly the same, except for the added \"@@, ## and entity label\" annotations for each entity detected.\nText span of an entity means the actual words/characters that form the entity in the text. An entity's span can contain single or multiple words but never partial words.\nThe format to follow while marking an entity with its label is @@<entity_text_span>##<label>\n\nPredefined Entity Categories (use these exact labels after ##):\n[\n   \"anatomical_location\", \"animal\", \"biomedical_technique\", \"bacteria\",\n   \"chemical\", \"dietary_supplement\", \"DDF\", \"drug\", \"food\", \"gene\",\n   \"human\", \"microbiome\", \"statistical_technique\"\n]\n\nNote - DDF stands for Disease, Disorder, or Finding. The remaining categories refer to their conventional or scientific meaning.\nAlso, If the first word or first set of words in output belong to an entity then ensure to start the output with @@ and follow rest of instructions.\n\n### Text\nInput: Analysis of the Efficacy of Diet and Short-Term Probiotic Intervention on Depressive Symptoms in Patients after Bariatric Surgery: A Randomized Double-Blind Placebo Controlled Pilot Study.\nOutput:"}, {"role": "assistant", "content": "Analysis of the Efficacy of Diet and Short-Term Probiotic Intervention on @@Depressive Symptoms##DDF in @@Patients##human after Bariatric Surgery: A Randomized Double-Blind Placebo Controlled Pilot Study."}]}

C. Fine-Tuning Details

While most hyperparameters were set to "auto" by the OpenAI API, the configuration for our best-performing model is summarized in Table 3.

Table 3
Fine-Tuning Configuration for GPT-4.1-mini.

Hyperparameter	Value
Base Model	gpt-4.1-mini
Number of Epochs	1
Batch Size	auto (resolved to 1)
Learning Rate Multiplier	auto (resolved to 1)
Training Set Size	638 examples (319 titles, 319 abstracts)