Do You Need a Frontier LLM, or Can Domain-Specific Encoders Compete in Gut–Brain Information Extraction?

Aleksis Datseris[1,2,*,†], Mario Kuzmanov[1,3,*,†], Maria Barouh[1,†], Chiara Barbero[5,†], Matilde Canelas[5,†], Margarida Ramos[5,†], Rute Costa[5,†] and Ivelina Nikolova-Koleva[1,4,*,†]

[1]*Ontotext, 111R Tsarigradsko Shosse blvd., 1784 Sofia, Bulgaria*
[2]*FMI, Sofia University, 5 "James Bourchier" Blvd., 1164 Sofia, Bulgaria*
[3]*Heidelberg University, Grabengasse 1, 69117 Heidelberg, Germany*
[4]*IICT, Bulgarian Academy of Sciences, Acad. G. Bonchev Str, bl.2, 1113 Sofia, Bulgaria*
[5]*NOVA University of Lisbon | CLUNL, Av. de Berna, 26 C, 1069-061, Lisboa, Portugal*

Abstract

This paper presents a comprehensive suite of Information Extraction systems developed for the GutBrainIE Task at the BioASQ Lab of CLEF 2026, aimed at elucidating the gut-brain interplay from biomedical literature. For the Named Entity Recognition (NER) subtask, we evaluate domain-specific BERT models, GLiNER architectures, and ensemble methods, incorporating distant supervision and data augmentation. Our GLiNER-based approach demonstrated superior robustness on unseen test data, securing second place. For Named Entity Recognition and Disambiguation (NERD), which we address as an entity linking problem, we propose a two-stage hybrid system—which also ranked second. It combines deterministic frequency-based gazetteers for in-vocabulary terms and tackles out-of-vocabulary disambiguation with PubMedBERT and a fine-tuned SapBERT embedding models. Finally, for Mention-level (M-RE) and Concept-level (C-RE) Relation Extraction, we contrast an encoder-based ATLOP architecture, supervised fine-tuning of an open-weights LLM (Qwen3.5-4B), and few-shot prompting of proprietary GPT-5 models (GPT-5.4 and GPT-5.5). Our Concept-level Relation Extraction system also ranked second in the respective subtask.

Keywords

NER, Entity Linking, Relation Extraction, Gut-Brain Interplay, Deep Learning

1. Introduction

The bidirectional communication between the central and the enteric nervous system, commonly referred to as the gut-brain axis, has emerged as a critical frontier in biomedical research. The growth of scientific literature exploring this interplay underscores the need for automated Information Extraction (IE) systems capable of accurately identifying and linking biomedical entities, as well as extracting their complex relationships. The GutBrainIE Task [1], part of the BioASQ Lab [2] at CLEF 2026[1], addresses this need by providing a benchmark across four interrelated subtasks: Named Entity Recognition (NER), Named Entity Recognition and Disambiguation (NERD)—which we approach as an entity linking (EL) problem—and both Mention-level (M-RE) and Concept-level (C-RE) Relation Extraction (RE).

In this paper, we present a comprehensive suite of methodologies developed to tackle the complexities of the GutBrainIE Task. We also provide insights into improving the developed approaches and the datasets used for system training. For the NER subtask, we explore domain-specific pre-trained language models and introduce highly robust approaches utilizing GLiNER [3] architectures with distant

CLEF 2026: Conference and Labs of the Evaluation Forum, September 21–24, 2026, Jena, Germany
*Corresponding author.
†These authors contributed equally.
aleksis.datseris@graphwise.ai (A. Datseris); mario.kuzmanov@graphwise.ai (M. Kuzmanov); maria.barouh@graphwise.ai (M. Barouh); chiarabarbero@fcsh.unl.pt (C. Barbero); amcanelas@fcsh.unl.pt (M. Canelas); mvramos@fcsh.unl.pt (M. Ramos); rute.costa@fcsh.unl.pt (R. Costa); ivelina.nikolova@graphwise.ai (I. Nikolova-Koleva)
0009-0001-7997-3942 (A. Datseris); 0009-0003-6320-6663 (M. Kuzmanov); 0000-0001-8120-1784 (C. Barbero); 0009-0001-5010-0849 (M. Canelas); 0000-0001-7209-3806 (M. Ramos); 0000-0002-3452-7228 (R. Costa); 0009-0006-6322-0256 (I. Nikolova-Koleva)

[1]https://clef2026.clef-initiative.eu/

supervision. We further apply other enhancements like weighted training and data augmentation. For the NERD subtask, we propose a novel two-stage entity linking pipeline that resolves the non-deterministic shortcomings of the proposed baseline; our system combines deterministic, frequency-based gazetteers for in-vocabulary terms with model voting using PubMedBERT[2] [4] and a fine-tuned SapBERT [5] model for out-of-vocabulary concept disambiguation. Finally, to address the RE subtasks, we conduct a broad comparative analysis benchmarking an efficient, encoder-based ATLOP [6] architecture against the supervised fine-tuning of open-weights large language models such as Qwen3.5-4B [7] and the few-shot prompting of proprietary GPT-5 models (GPT-5.4 and GPT-5.5).

Our top-performing pipelines achieved highly competitive results on the official leaderboards, ranking 2nd in the NER, NERD and Concept-level RE (C-RE) subtasks, while our Mention-level RE (M-RE) system ranked 5th. Through our extensive experiments and error analyses, we demonstrate that while massive generative models excel at complex relation reasoning, specialized architectures combined with task-tailored heuristics remain exceptionally effective and resource-efficient for foundational IE tasks.

2. Related Work

Similarly to the general-domain NER, early approaches to Biomedical Named Entity Recognition (BioNER) relied heavily on rule-based systems and conditional random fields. With the advent of transformer architectures, domain-specific pre-trained language models (PLMs) such as PubMedBERT [4] and BioBERT [8] became the de facto standard due to their ability to capture complex biomedical language patterns. Recently, the Generalist and Lightweight Information Extraction (GLiNER) architecture has shown immense promise for open-vocabulary BioNER. For instance, the GLiNER-BioMed [9] suite adapts this architecture using large language models for synthetic data distillation, achieving state-of-the-art zero-shot and few-shot capabilities across numerous biomedical datasets. Additional studies have highlighted that GLiNER frameworks, when coupled with targeted gazetteer-based post-processing, can substantially mitigate common misclassifications in highly specialized domains [10].

Biomedical Entity Linking is the task of mapping extracted entity mentions to standardized identifiers within biomedical knowledge bases. Historically, this was tackled using strict gazetteers or frequency matching. To address the pervasive issue of synonymy and textual variation, modern deep learning approaches have shifted toward embedding-based semantic similarity and both approaches are often used in combination. SapBERT is a prominent pretraining scheme that self-aligns the representation space of biomedical entities, offering a robust "one-model-for-all" solution that achieves state-of-the-art performance in medical entity linking, even without task-specific supervision [5].

While sentence-level RE has seen considerable success, many complex relationships—particularly in topics like the gut-brain axis—span multiple sentences, necessitating document-level reasoning [11] [12]. To address this, methods like Adaptive Thresholding and Localized Context Pooling (ATLOP) [6] have been introduced to better capture the correlations between entities across a document. ATLOP is frequently used as a robust backbone for document-level RE frameworks, including those prioritizing self-distillation and resource efficiency [13].

The integration of Large Language Models (LLMs) has fundamentally shifted RE and IE paradigms. Both supervised fine-tuning (e.g., via LoRA) and in-context learning (few-shot prompting) allow models to handle overloaded predicates and infer complex document-level relations with high precision, complementing the robust foundational capabilities of encoder architectures like ATLOP.

3. Tasks and Data

The GutBrainIE Task [1] is organised within the BioASQ Lab [2] of CLEF 2026. It aims at fostering the development of IE systems that help biomedical scientists better understand the gut-brain interplay. Within the Task, there are four subtasks:

[2]https://huggingface.co/NeuML/pubmedbert-base-embeddings

- **Named Entity Recognition (NER, Subtask 6.1.1)** - the goal is to classify a mention of interest into one of the 13 pre-defined categories (see Appendix F). Furthermore, a valid entity (E) is defined as the mention span boundaries (inclusive), the location – either found in title or abstract, and the label/category:

```
E = {"start_idx": 75, "end_idx": 82, "location": "title", "text_span": "patients", "label": "human" }
```

- **Named Entity Recognition and Disambiguation (NERD, Subtask 6.1.2)** - the goal is to extend the identified entity by linking it to a resource from a pre-defined knowledge base:

```
{"uri": "http://id.nlm.nih.gov/mesh/D010361"}
```

- **Mention-level Relation Extraction (M-RE, Subtask 6.2.1)** - the goal is to identify the type of association between entity mentions:

```
{
    "subject_text_span": "intestinal microbiome",
    "subject_label": "microbiome",
    "predicate": "located in",
    "object_text_span": "patients",
    "object_label": "human"
}
```

- **Concept-level Relation Extraction (C-RE, Subtask 6.2.2)** - extension of the previous subtask to the concept level. Now, relations between linked concepts are to be classified:

```
{
    "subject_uri": "http://purl.obolibrary.org/obo/NCIT_C93019",
    "subject_label": "microbiome",
    "predicate": "located in",
    "object_uri": "http://id.nlm.nih.gov/mesh/D010361",
    "object_label": "human"
}
```

The organisers provide a total of 5001 annotated PubMed abstracts. Depending on the quality of their annotation, they are split into Gold, Silver, and Bronze corpora. Refer to Table 1 for an overview of each corpus. Detailed descriptions of the GutBrainIE Task and data are provided by Martinelli et al. [1]. Throughout this paper, the term "label" carries a dual meaning. Within the annotated data, it denotes the entity category; however, in the context of the NERD experiments, it is used interchangeably with an entity's "name" to represent its alternative surface forms.

4. Named Entity Recognition Approaches

This section details our NER approaches. It starts by presenting the functional formulation of the problem, then motivates the data techniques considered and concludes with a thorough overview of the experimental groups. Our best NER approach ranks second in the final leaderboard.

4.1. Task Formulation

We stick to the standard approach for solving NER as a token classification task, i.e., labeling each token in an input sequence. To do so, each example is converted into a BIO format representation (Ramshaw and Marcus [14]). `B` is used to mark a token that indicates the start of an entity, while `I` marks every next token that is part of it. `O` is reserved for tokens that are outside spans of interest. For example, if we have a single label *DDF*, then our model would need to choose between *B-DDF, I-DDF, O* for each token. Finally, tokens are aligned and mapped back into entities on the basis of their labels and offsets to produce the predicted entity mentions. However, token classification is not the only possible approach. Other solutions utilize span-representation modules, but still rely on the meaning of each token in latent space to form a final representation for the entity. For the NER subtask, we do not fine-tune large (in the modern sense) LLMs. This is partially motivated by our hardware setup, specified in Appendix I.

Table 1
Abbreviations used for the different data sources with their corresponding weights for training. **C-RE** refers to the annotations for the Concept-level Relation Extraction subtask.

Abbreviation	Corpus	Weight	Description
G	Gold	1.5	639 articles. Expert-curated.
S	Silver	1.0	811 articles. Mid-quality, student annotated.
S25	Silver from GutBrainIE 2025 [18]	1.0	499 articles. Same as Silver, automatic C-RE annotations.
B	Bronze	1.0	2972 articles. Automatic annotations using baseline models.
BA	Re-annotated Bronze	1.0	2972 articles. Automatic annotations using our ensemble (only for NER).
A	Augmented	1.0	6720 articles. Same annotation procedure as in BA.
dev	Development set	1.5	80 articles.
test	Test set (no access to annotations)	–	80 articles.

4.2. Data Techniques

We apply three distinct approaches, which are also visible in Figures 1–2.

- **Re-annotation** - the original Bronze standard data is annotated through distant supervision with a fine-tuned GLiNER-based model[3], which is also the baseline system for the NER subtask. However, as we find that our models consistently get higher scores on the dev set than the baseline, it is suboptimal to use the same annotations. Therefore, inspired by the preliminary success of the ensemble approaches, we form an ensemble based on the most successful **BERT** experimental groups and re-annotate all articles in the Bronze data. The **BERT** and **Ensemble** experiments are described in Section 4.3. The ensemble used for annotation of the Bronze data is always the same - **Ensemble-Experiment-1-A** (refer to Figure 3 and Subsection 4.3.3). Note, only the Bronze corpus is re-annotated. We are inspired by the work of Han et al. [15], which follows a similar technique and shows that it contributes to winning the NER subtask from last year. They also show that input formatting can be optimized to slightly improve overall performance. We apply their formatting strategy to all data points for the **BERT** group. Namely, the input becomes:

$$[\text{CLS}]\,[\text{TITLE}]\,t_1,\ldots,t_m\,[\text{ABSTRACT}]\,a_1,\ldots,a_n\,[\text{PAD}]$$

  where $[\text{TITLE}]$ and $[\text{ABSTRACT}]$ are newly learned tokens.
- **Weighted Training** - refers to the process of multiplying the final loss by a pre-defined small number (weight) that accounts for how significant the current example should affect training. Such an approach proved beneficial in Andersen et al [16]. They experiment with various weighted combinations and show how this directly impacts the performance. We choose to use the weights shown in Table 1.
- **Data Augmentation** - finally, we augment the corpora with 6720 related articles fetched directly through the PubMed API. We use the same dataset prepared and described in Datseris et al. [17] (Augmented Data). Although we originally mention 6728 articles extracted, 8 are omitted due to missing annotations.

Furthermore, to keep the definitions of the experimental groups compact, we introduce an abbreviation vocabulary. Specifically, Table 1 shows how we refer to the different data splits in the next section.

4.3. Experimental Groups

Our approaches mainly focus on fine-tuning domain-specific PLMs, and complementarily apply a majority-voting technique to leverage the capabilities of model ensembles. For ease of reference, we define each experiment with a unique name, which corresponds directly to the results in Section 5.

[3]https://huggingface.co/numind/NuNER_Zero

4.3.1. BERT

We call our first experimental group **BERT**. The following open-source BERT-based models [19] are considered for the experiments:

- `microsoft/BiomedNLP-BiomedBERT-base-uncased-abstract-fulltext`[4] [20] - we refer to as `PubMedBERT-s` throughout the paper. The model comes with an MIT license and has only about 110M parameters, which is orders of magnitude smaller than current LLMs. It closely follows the standard BERT architecture and uses Masked Language Modeling (MLM), and Next Sentence Prediction (NSP) as objectives during pre-training. We find `PubMedBERT-s` particularly useful for the GutBrainIE NER subtask because (1) it is pre-trained from scratch on PubMed[5] abstracts and PubMedCentral[6] full-texts, which presupposes a much better fit to the biomedical vocabulary. (2) The model consistently ranks top on various biomedical language understanding and reasoning benchmarks [7].
- `microsoft/BiomedNLP-BiomedBERT-large-uncased-abstract`[8] - we decide to employ `PubMedBERT-l`, which is released by the same authors as `PubMedBERT-s` and, thus, shares the same license. However, this version has a larger encoder network size (335M parameters) and, in contrast, is pre-trained on only PubMed abstracts. Otherwise, the model remains structurally similar to `PubMedBERT-s`.
- `dmis-lab/biobert-large-cased-v1.1`[9] [8] - we refer to as `BioBERT`. `BioBERT` is released under Apache 2.0 License with about 364M parameters, making it the largest model in the **BERT** experimental group. However, no changes to the original BERT architecture nor pre-training objectives are made. While the corpora of the model are dominated by domain-specific vocabulary, the authors keep the original English Wikipedia and BooksCorpus datasets. Our specific `BioBERT` checkpoint is augmented with a large number of PubMed abstracts but no full-texts.
- `sultan/BioM-ELECTRA-Large-Discriminator`[10] [21] is based on the ELECTRA architecture [22]. ELECTRA-based models require significantly less compute resources while retaining similar or improved performance over their BERT-based counterparts. However, they still keep the full standard attention mechanism, therefore limiting the length of input sequences to the standard 512 tokens. The key difference is that the MLM objective is replaced with token detection. Namely, during training, the weights of two different neural networks are updated - one is used to corrupt some tokens with plausible alternatives sampled from the other, which has the function to generate them. The goal is to train a discriminative model that classifies each corrupted token as either generated by the network or not. For our downstream task, we only fine-tune the discriminator network, which is the pre-trained ELECTRA model. We refer to it as `BioELECTRA` in our experiments for simplicity. This and the other models from the biomedical transformers family developed by the authors are available under an Apache 2.0 License. `BioELECTRA` is pre-trained on PubMed abstracts using the same vocabulary as `PubMedBERT-l`, with a total of 332M parameters.

The models from the **BERT** group are combined into 8 different computational experiments. Figure 1 gives an overview of each experiment and defines the names used in Section 5. We use the **Backbone** to get the last hidden state of the encoder model for the sequence. The outputs are passed through a dropout layer for better regularization and transformed into emission probabilities through a linear dense layer (Dense). On top of the architecture, a CRF layer decodes the final BIO predictions using the Viterbi algorithm [23]. During training, our goal is to minimize the loss, which is the negative

[4]https://huggingface.co/microsoft/BiomedNLP-BiomedBERT-base-uncased-abstract-fulltext
[5]https://pubmed.ncbi.nlm.nih.gov/
[6]https://pmc.ncbi.nlm.nih.gov/
[7]https://microsoft.github.io/BLURB/
[8]https://huggingface.co/microsoft/BiomedNLP-BiomedBERT-large-uncased-abstract
[9]https://huggingface.co/dmis-lab/biobert-large-cased-v1.1
[10]https://huggingface.co/sultan/BioM-ELECTRA-Large-Discriminator

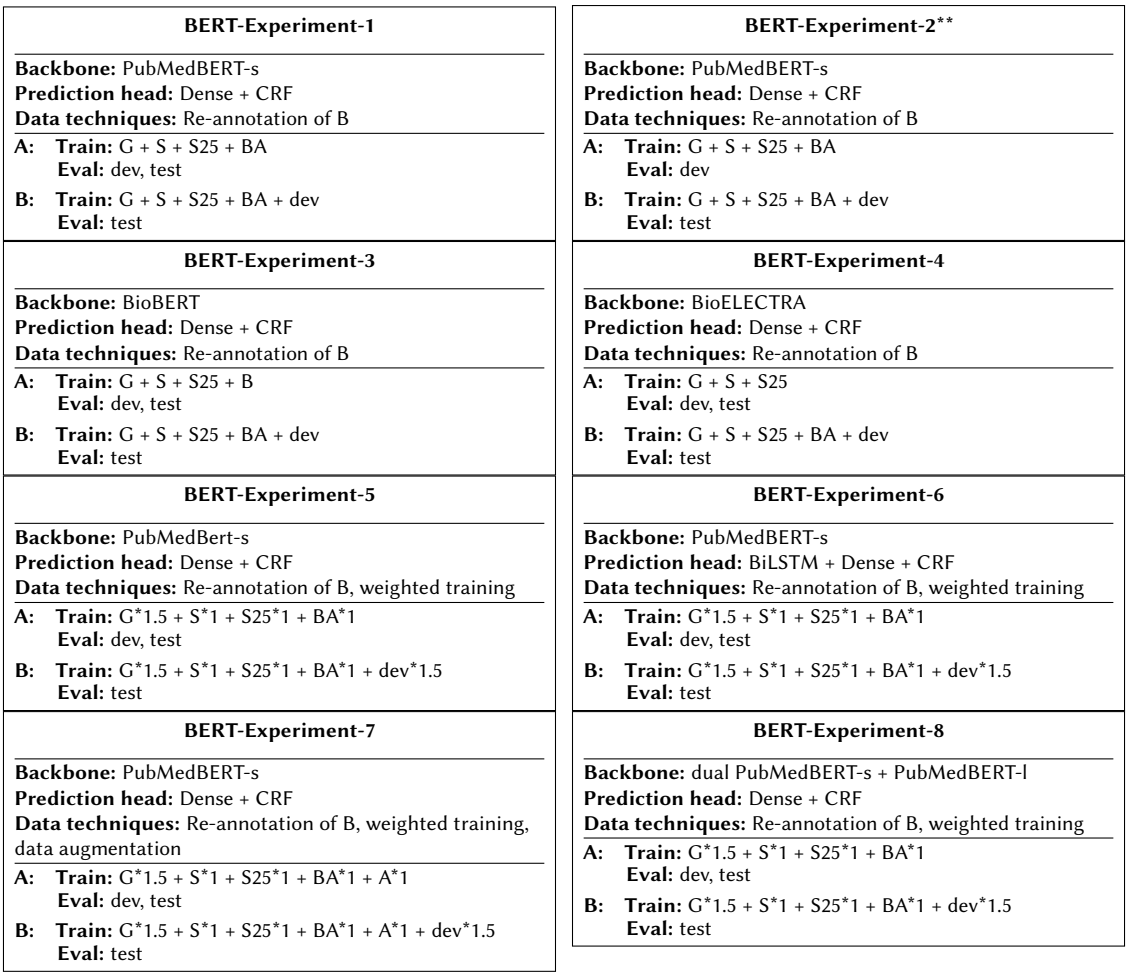

Figure 1: Overview of the **BERT** experiments. Each experiment uses a common backbone, prediction head, and data-technique setting, with two train/eval variants shown inside each box. Precisely, the main difference between variants is that, for the second variant, we also train on the dev set. Otherwise, dev is used for evaluation. Although **BERT-Experiment-1** and **BERT-Experiment-2** have identical cards, the latter one is starred because it marks the point at which we start using different learning rates for the backbone and the prediction head's components.

log likelihood of the predicted label sequence given the emission scores, through backpropagation. Starting from **BERT-Experiment-5**, we additionally multiply the loss value by a pre-determined weight to incorporate the annotation quality into the magnitude of the result. Refer to Appendix E for all hyperparameters of the experiments. All models run for 20-25 epochs with 3 epochs patience. The weights for weighted training are also hyperparameters, shown in Table 1. However, we have some deviations from the described process. For example, **BERT-Experiment-6** passes an output additionally encoded with a single-layered bidirectional LSTM (Vennerod et al. [24]) to the Dense layer. The BiLSTM representations go through a GELU activation [25] first, before the linear transformation turns them into emission scores. Finally, in **BERT-Experiment-8** we employ a dual encoder architecture in which two models with very similar vocabularies and domains are trained jointly. Unlike other methods, now we get two input sequence representations that go through a dropout layer. Then, we use two different Dense layers to produce the emission matrix, which is based on the logit fusion controlled by a learnable α weight:

$$w = \text{sigmoid}(\alpha),$$
$$\mathbf{E} = w\mathbf{E}_1 + (1 - w)\mathbf{E}_2,$$

where $\mathbf{E}_1$ and $\mathbf{E}_2$ denote the emission scores produced by the linear layers of the first and second models, respectively.

4.3.2. GLiNER

Our next group refers to all experiments related to GLiNER-based models [3] and is therefore called **GLiNER**. The GLiNER architecture builds upon a small bidirectional Transformer-based model (BERT or DeBERTa [26]) with two consequent parallel layers trying to match the input entity label with its corresponding span representation. Namely, GLiNER encodes an input "prompt" with entity labels separated by a special token `[ENT]`, which is concatenated (with `[SEP]`) to the input sequence. The high-dimensional vector representations are passed either to a Feed-Forward Neural Network (FNN) layer if they correspond to the label, or to a span representation layer otherwise. A simple matching score based on the dot product and sigmoid activation is computed for each label-span pair. If the calculated score passes a certain threshold, the span receives its corresponding label. Most commonly, GLiNER is used as a span-based model, however, the authors' baseline utilizes an architecture suited for the standard token classification task.

Specifically, the checkpoints that we include are listed as follows:

- `Ihor/gliner-biomed-large-v1.0`[11] [27] - released under Apache 2.0 License, we further refer to this model as `GLiNER-Biomed`. `GLiNER-Biomed` has about 478M parameters. It is part of the suite for efficient models for open biomedical NER (GLiNER-Biomed), which are pre-trained on biomedical domain synthetic data from PubMed articles. This is also the largest model we use within this experimental group and in general for the NER subtask.
- `gliner-community/gliner_large-v2.5`[12] shares the same Apache license and has 459M parameters. However, the model is not specialized for the biomedical domain but instead pre-trained on the synthetic PileNER corpus. For simplicity, we call this checkpoint `GLiNER-V2.5` in our experiments.
- `Ihor/gliner-biomed-bi-base-v1.0`[13] - this checkpoint falls under the same group as `GLiNER-Biomed`. However, it has a smaller size (242M parameters) and a dual independent encoder architecture. One encoder has the function of extracting the vector representations of entity labels, while the other encodes the text. Thus, the original FFN and span-representation modules become much more expressive. However, the rest of the computation to match the label-span pairs stays unchanged from the standard GLiNER. The main bottleneck of such approaches that are built upon BERT-like models remains the expensive attention computation, which scales memory and compute requirements quadratically over time (i.e. increasing sequence length). We use the name `GLiNER-BiBiomed` for further mentions of this model.
- `numind/NuNER_Zero-span`[14] [28] - which we refer to as `GLiNER-span`. Note that this is the span-based version of the baseline model provided by the organizers. They use a fine-tuned `numind/NuNER_Zero`, which is trained on the NuNER corpus[15] and is shown to generally perform better than its standard GLiNER counterpart but slightly worse than `GLiNER-span`. However, unlike `GLiNER-span`, which cannot detect entities spanning more than 12 tokens, the baseline model does not have an entity token limit size. Both models are under MIT license and with 449M parameters, but we only report the experiments with `GLiNER-span`.

In total, we conduct 4 different computational experiments with the models in the **GLiNER** group. They are illustrated in Figure 2 with the confidence thresholds used for predictions, and the data techniques. We empirically determine the optimal threshold for each model by performing inference runs on the dev set. Only the best values are reported and no weighted training is applied. To fine-tune the models, we convert the data to the suitable NER format and use the GLiNER framework, which is available on PyPI[16]. We adapt the scripts for data conversion provided by the authors[17] to match the

[11] https://huggingface.co/Ihor/gliner-biomed-large-v1.0
[12] https://huggingface.co/gliner-community/gliner_large-v2.5
[13] https://huggingface.co/Ihor/gliner-biomed-bi-base-v1.0
[14] https://huggingface.co/numind/NuNER_Zero-span
[15] https://huggingface.co/numind/NuNER-v2.0
[16] https://pypi.org/project/gliner/
[17] https://github.com/MMartinelli-hub/GutBrainIE_2026_Baseline

GLiNER-Experiment-1	GLiNER-Experiment-2**
Backbone: GLiNER-V2.5 **Threshold & Data Techniques:** 0.6 & Re-annotation of B **Train:** G + S + S25 + BA **Eval:** dev, test	**Backbone:** GLiNER-Biomed **Threshold & Data Techniques:** 0.7 & Re-annotation of B **Train:** G + S + S25 + BA **Eval:** dev, test
GLiNER-Experiment-3	**GLiNER-Experiment-4**
Backbone: GLiNER-BiBiomed **Threshold & Data Techniques:** 0.7 & Re-annotation of B **Train:** G + S + S25 + BA **Eval:** dev, test	**Backbone:** GLiNER-span **Threshold & Data Techniques:** 0.7 & Re-annotation of B **Train:** G + S + S25 + BA **Eval:** dev, test

Figure 2: Overview of the **GLiNER** experiments. **GLiNER-Experiment-2** is starred because it achieved 2nd place on the official test set of the NER subtask. Unlike the train/eval variants in Figure 1, here the dev set is only used for evaluation.

Ensemble-Experiment-1	Ensemble-Experiment-2
Ensemble rule: Majority voting (3/4 votes)	**Ensemble rule:** Majority voting (3/4 votes)
A: **Experiments:** BERT-Exp. 3-A, BERT-Exp. -4-A, BERT-Exp. -7-A, BERT-Exp. -8-A **Eval:** dev, test	**A:** **Experiments:** BERT-Exp. -2-A, BERT-Exp. -7-A, BERT-Exp. -8-A, GLiNER-Exp. -2 **Eval:** dev, test
B: **Experiments:** BERT-Exp. -3-B, BERT-Exp. -4-B, BERT-Exp. -7-B, BERT-Exp. -8-B **Eval:** test	**B:** **Experiments:** BERT-Exp. -2-B, BERT-Exp. -7-B, BERT-Exp. -8-B, GLiNER-Exp. -2 **Eval:** test

Figure 3: Overview of the **ensemble** experiments. Each ensemble combines four previously defined computational experiments. Note, for experiment variant **B** we use the models trained on the dev set, otherwise dev is used for evaluation.

span-based input, or directly re-use them for the token classification experiments. The input sequence length is kept at 512 tokens for all models. In addition, in Appendix E the hyperparameters used during the fine-tuning process are reported.

4.3.3. Ensemble

Finally, our **Ensemble** experimental group is meant to combine any subset of systems through majority voting. For a given entity boundary corresponding to a text span, we consider a "vote" as the label that each model assigns for this boundary. Thus, our approach addresses label conflicts but misses potential boundary mismatches or span overlaps. We first determine the least number of votes needed to include an entity, i.e., majority threshold: $threshold = (\frac{all\_votes}{2}) + 1$. Then, for each qualifying entity (that is, predicted 3 out of 4 times), we choose its most common label from the models' predictions. Following the procedure, we form the following ensemble groups:

- **Ensemble-Experiment-1** - includes two variants with four models exclusively taken from the **BERT** experimental group.
- **Ensemble-Experiment-2** - constitutes a mixture of the **BERT** and **GLiNER** groups, again limited to two groups with four distinct models.

Refer to Figure 3 for an overview of the **Ensemble** group.

5. Named Entity Recognition Results

Next, we present and discuss the results of the experiments described in Section 4. Our best NER approach (**GLiNER-Experiment-2**) ranks 2nd on the final leaderboard with a Micro-F1 score of 84.23% on the dev and 83.85% on the test set. Furthermore, all of our submitted runs beat the baseline results.

Table 2
All NER experimental results. Columns report macro-averaged F_1 (Ma-F_1) and micro-averaged precision, recall, and F_1 (μ-P, μ-R, μ-F_1) on the dev and test sets. Refer to Figures 1–3 for an overview of the evaluated methods. The unfilled cells indicate that the system is not evaluated on the corresponding split. Best values per column are shown in bold. **BASELINE** denotes the authors' challenge baseline for NER.

	Dev				Test			
Model	Ma-F_1	μ-P	μ-R	μ-F_1	Ma-F_1	μ-P	μ-R	μ-F_1
BASELINE	74.38	77.10	82.78	79.84	72.67	77.82	82.21	79.96
BERT Group								
BERT-Experiment-1-A	75.74	82.52	82.94	82.73	73.91	81.95	81.76	81.86
BERT-Experiment-1-B	–	–	–	–	73.14	79.58	82.17	80.85
BERT-Experiment-2-A	77.75	83.52	83.42	83.47	73.14	82.38	80.95	81.66
BERT-Experiment-2-B	–	–	–	–	74.61	81.72	82.51	82.11
BERT-Experiment-3-A	78.57	80.40	83.97	82.15	73.93	80.51	81.76	81.13
BERT-Experiment-3-B	–	–	–	–	71.50	82.03	79.69	80.84
BERT-Experiment-4-A	74.52	81.14	83.97	82.53	71.01	80.29	81.25	80.77
BERT-Experiment-4-B	–	–	–	–	70.96	81.26	82.47	81.86
BERT-Experiment-5-A	77.75	83.00	84.25	83.62	75.04	82.65	82.65	82.65
BERT-Experiment-5-B	–	–	–	–	74.61	81.38	82.77	82.07
BERT-Experiment-6-A	77.20	82.86	83.78	83.31	74.27	82.66	81.65	82.16
BERT-Experiment-6-B	–	–	–	–	73.61	79.72	82.32	80.99
BERT-Experiment-7-A	78.49	83.46	85.44	84.44	74.07	81.57	82.84	82.19
BERT-Experiment-7-B	–	–	–	–	74.88	81.64	82.06	81.85
BERT-Experiment-8-A	78.94	82.31	**85.48**	83.87	74.73	81.48	83.02	82.25
BERT-Experiment-8-B	–	–	–	–	75.44	82.20	83.17	82.68
GLiNER Group								
GLiNER-Experiment-1	77.71	82.90	84.05	83.47	76.19	83.33	83.36	83.34
GLiNER-Experiment-2	78.76	86.15	82.39	84.23	**76.65**	**86.13**	81.69	**83.85**
GLiNER-Experiment-3	77.88	**86.34**	81.71	83.96	74.33	86.07	81.06	83.49
GLiNER-Experiment-4	77.05	81.84	83.82	82.81	75.17	81.96	**83.84**	82.89
Ensemble Group								
Ensemble-Experiment-1-A	80.06	85.46	85.12	85.29	74.80	83.40	82.32	82.86
Ensemble-Experiment-1-B	–	–	–	–	74.32	83.58	81.50	82.53
Ensemble-Experiment-2-A	**80.34**	85.69	85.01	**85.34**	75.47	84.49	81.76	83.10
Ensemble-Experiment-2-B	–	–	–	–	75.71	84.20	81.20	82.99

Dev set. We get the best micro-F1 performance (84.44%) in Experiment-7 from the **BERT** group (Table 2). That is, employing `PubMedBERT-s` as backbone and incorporating all available data techniques. From the **GLiNER** group, Experiment-2 is the leader with 84.23%. Finally, the best performing group is the **Ensemble-Experiment-2** achieving 85.34%.

Test set. The **GLiNER** group notably gets almost the same results as in the dev set. This speaks positively about the generalization and robustness capabilities of GLiNER for NER in general. `GLiNER-Biomed` trained on all provided corpora, and with included pre-annotated Bronze corpus dominates with 83.85% Micro-F1. In the **BERT** group, a larger drop in performance is observed. The most prominent example is in Experiment-7-A, where the Micro-F1 drops from 84.44% on the dev, to 82.19% on the test, despite the fine-tuning with augmented data. Our dual encoder architecture in Experiment-8 performs the best (82.68%) showing the potential of joint model fine-tuning and weighted training. For the **Ensemble** group, the differences between dev and test time are the most obvious. While leading the dev set with Micro-F1 scores of 85.29% and 85.34%, they drop to 82.86% and 83.10% on test time for Experiment-1-A and Experiment-2-A, respectively.

5.1. Named Entity Recognition Error Analysis

We conducted a qualitative analysis of the false positive (FP) and false negative (FN) errors obtained by comparing the expert annotations with the model predictions on the development set. The analysis

consisted of manually examining recurrent discrepancies between the gold-standard annotations and the automatically predicted entities in order to identify common error patterns.

The most frequent discrepancies were associated with annotation inconsistency, entity boundary variation, semantic ambiguity, and inconsistent handling of abbreviations and repeated mentions. In several cases, semantically plausible predictions were classified as false positives or false negatives due to differences in annotation granularity or span delimitation across the corpus. For example, entities such as "intestinal barrier" were annotated interchangeably with the labels *microbiome* and *anatomical location* in different contexts, while abbreviated mentions such as "NC", "SCD", and "aMCI" were inconsistently annotated across repeated occurrences within the same abstract.

Boundary-related discrepancies also represented a major source of error. The model frequently identified the core biomedical concept while omitting preceding modifiers, for example predicting "microbiota dysbiosis" instead of the gold-standard span "preexisting microbiota dysbiosis". Similar mismatches were observed in coordinated expressions such as "pre- and probiotics", where semantically plausible predictions diverged from the span boundaries defined in the gold standard.

Additional errors were associated with semantic ambiguity and formatting irregularities. Broad lexical items such as "host" occasionally generated semantically underspecified predictions, whereas formatting irregularities involving concatenated tokens, such as "BackgroundAlzheimer's disease", introduced segmentation difficulties that negatively affected entity recognition.

6. Named Entity Recognition and Disambiguation Approaches

The Named Entity Recognition and Disambiguation (NERD) subtask, which we address as an entity linking (EL) problem, is described in detail by Martinelli et al. [1]. This section outlines the methodologies, external resources, and experiments conducted during our participation. Notably, our top-performing system configuration achieved second place in the NERD subtask.

NER involves identifying entity spans within a given text and classifying them into predefined categories. In this work, NER is followed by entity linking (EL), which maps each extracted entity to a unique Uniform Resource Identifier (URI) within a target Knowledge Base (KB) provided by the organizers (a set of 3598 URIs). The sequence of NER and EL is referred to as NERD. Recent studies in biomedical Natural Language Processing (NLP) demonstrate that gazetteer-based approaches for NERD can outperform complex deep learning architectures. A notable example is the SNOMED challenge 2024[18] where the winning systems relied on such methods[19]. Motivated by these findings and the baseline approaches shared by the organizers[20], this paper proposes a two-stage approach for entity linking. First, entities present in the training data are resolved using a set of gazetteers. Subsequently, for out-of-vocabulary entities, embedding similarity is employed to identify the most analogous concept within the knowledge base.

6.1. Gazetteers

Two distinct gazetteers are used to match in-vocabulary entities from both the training and development sets [21]. This approach relies on the premise that in a highly constrained domain—such as biomedical abstracts concerning Gut-Brain Interplay—terms are rarely used interchangeably and entity ambiguity is inherently low. Consequently, entities that exactly match a gazetteer term have a high probability of mapping to the term corresponding URI.

[18] https://www.snomed.org/entity-linking-challenge
[19] https://github.com/drivendataorg/snomed-ct-entity-linking/tree/main/1st%20Place
[20] https://github.com/MMartinelli-hub/GutBrainIE_2026_Baseline
[21] During the dev phase, the development set was excluded from the gazetteers

6.1.1. Entity-Category Frequency Gazetteer

To leverage the output of the NER step, the organizers' baseline approach suggests to first generate a gazetteer from all training data entities (gold, silver, silver-2025 and dev datasets) including the entity text span, its category and a set of URIs. The URIs are collected in a set because a tuple $(text\_span, category)$ may occur in the corpus with more than one URI. The data is collected in a dictionary[22] with the following structure.

$$< text\_span, category >: set(URI\_1, URI\_2, ...)$$

The keys of the dictionary are formulated as tuples comprising the entity text span and its corresponding category. This formulation mitigates the ambiguity of entities that appear under multiple categories, ensuring that the assigned URIs are conditioned on both the textual mention and its semantic type. However this approach has its downsides because the values of the dictionary are sets and all URIs are treated equally. Then for entities with more than one URI, a random one is pooled from the set. This approach is non-deterministic and also not optimal from an accuracy point of view. To improve upon this, our solution uses the following structure:

$$< text\_span, category >: most\_frequent\_URI$$

Where $most\_frequent\_URI$ is the URI for the given entity and given category which occurs most often in the training dataset.

This primary gazetteer operates with the highest confidence score, requiring an exact match for both the entity's text span and its assigned category. This gazetteer will be referred later on as Entity-Category Frequency Gazetteer. However, this strict criterion is insufficient to fully resolve the EL task. In unseen texts, such as the test set, an entity may share a lexical span with a gazetteer entry but belong to a different category, resulting in a key mismatch. To address these cases, a secondary gazetteer is employed.

6.1.2. Entity Frequency Gazetteer

This secondary gazetteer aims to resolve the EL task for in-vocabulary entities that fail to satisfy the category constraint of the primary gazetteer. Specifically, it addresses cases where an entity's lexical span is present in the training data, but its currently assigned category differs from those observed in the training dataset. The organisers' baseline suggests the following structure for this gazetteer:

$$text\_span : set(URI\_1, URI\_2, ...)$$

However, again having a set as a value leads to non-deterministic behavior because this approach pops a random URI as result. To ignore the non-determinism and increase the certainty of assigning a given URI, the proposed approach uses the following dictionary structure:

$$text\_span : most\_frequent\_URI$$

Similarly to the Entity-Category Frequency Gazetteer, here $most\_frequent\_URI$ is the URI for the given entity which occurs most often in the training dataset.

These two gazetteers cover in-vocabulary entities with very high confidence. They are applied sequentially on the input entity (see Table 3) and if an exact match is found with a given key, the corresponding value (URI) is returned as result.

6.2. Embedding Similarity

The next steps of the approach deal with the out-of-vocabulary entities. They include usage of transformer models' embeddings to search for the most appropriate concept/URI in an indexed KB.

[22] By dictionary, we consider a data structure holding <key:value> pairs which is created at the pre-processing step.

6.2.1. Baseline

The task baseline approach uses `NeuML/pubmedbert-base-embeddings`[23] model for producing the embeddings. In the preprocessing stage, the model is used to build an embeddings index including the labels of all allowed concepts. These labels are collected from the training data (all entity mentions in the corpus) and also from external sources like UMLS [29]. At the EL stage, the model produces an embedding for the input entity, then the KB is searched for the most similar entry based on this embedding. The index of the returned entity corresponds to a concept URI which is considered as a closest match and is returned as result.

6.2.2. SapBERT EL model fine-tuning

To outperform the baseline, we train a SapBERT model using a dictionary generated with GPT-5-mini, containing preferred and alternative labels for each allowed URI.

URI-to-Label Generated Dictionary This dictionary contains a preferred label/name and a set of alternative labels/names for each allowed URI. In order to obtain these labels/names, we produce a short definition for each concept, using the context in which the concept appears in the training set. The set of contexts is sent to the model which generates the definitions. These definitions along with the URIs are then used for a second query which instructs the model to return a preferred label/name and a list of alternative labels/names for each concept. The structure of the resulting dictionary is demonstrated using the concept of "Probiotics" as an example.

```
"http://purl.obolibrary.org/obo/NCIT_C93144": {
    "pref_label":"Probiotics",
    "alt_labels":["Beneficial microbes",
        "Live biotherapeutic product",
        "Live cultures",
        "Live microbial supplement",
        "Probiotic",
        "Probiotic microorganisms",
        ...]
    }
```

At the next step, the entity names in this dictionary are joined with the Entity-Category Gazetteer and used as a training set for fine-tuning a SapBERT EL model. The KB is then indexed using this model. To improve the search precision during inference, the KB index is partitioned by entity category. At inference time, an embedding of the input entity is generated and utilized to query the vector index of the corresponding category for the most similar entry. Ultimately, the concept identifier associated with this retrieved entry is returned as the final result.

The orchestration of the above described gazetteers, models and indices is illustrated in Appendix G.

6.3. Ensemble Systems

Two distinct ensemble systems are evaluated during development. Their architectures are outlined in Table 3. The gazetteers process in-vocabulary terms; if a match for the input entity is identified, the pipeline terminates, and the result is output. This early termination can occur after either Step 1 or Step 2. If no match is found, it indicates that the entity is out-of-vocabulary (unseen in the training data). In Step 3, these unseen entities are forwarded to one (EL Ensemble A) or more transformer models (EL Ensemble B) to generate embeddings. If one model is employed, the most similar entity is returned as result. Otherwise, the entity with higher similarity is returned.

The baseline system is very similar to EL Ensemble A, with the exception that it employs non-deterministic gazetteers rather than frequency gazetteers. In the baseline implementation, an entity is mapped to a set of URIs if it is associated with multiple URIs in the training data. Consequently, upon a

[23] https://huggingface.co/NeuML/pubmedbert-base-embeddings

successful match, a URI is selected at random from this associated set and returned. In contrast, the frequency gazetteers strictly return the single URI most frequently linked to that entity.

Table 3
Entity Linking ensemble systems workflow

	Step	**EL Ensemble A**	**EL Ensemble B**
In-vocabulary terms steps	1	Apply Entity-Category Frequency Gazetteer If a match is found Return the corresponding concept URI	Apply Entity-Category Frequency Gazetteer If a match is found Return the corresponding concept URI
	2	Apply Entity Frequency Gazetteer If a match is found Return the corresponding concept URI	Apply Entity Frequency Gazetteer If a match is found Return the corresponding concept URI
Out-of-vocabulary terms step	3	Embed the entity with PubMedBERT model Search for similar entities in KB Return the concept URI associated to the entity with highest similarity	Embed the entity with SapBERT model Search for similar entities in KB Save the entity with highest similarity Embed the entity with PubMedBERT model Search for similar entities Save the entity with highest similarity Compare the confidence scores of both entities and return the URI associated with the higher-scoring entity.

7. Named Entity Recognition and Disambiguation Results

Entity linking performance depends not only on the efficacy of the linking approach itself but also on the quality of the input Named Entity Recognition annotations. In Table 4, we present the evaluation details utilizing the outputs from the top three NER systems during the development (column **Dev**). They turned out to be among the best ones on the Test set as well (see column **Test**) and the system **EL Ensemble B** with input from **GLiNER-Experiment-2** ranked second in the GutBrainIE Task achieving μ-F_1 of 0.61 on unseen data. The results show that both EL ensembles strictly depend on their input and follow the same trend as the underlying NER systems, scoring μ-F_1 on the dev set as follows: **Ensemble-Experiment-2-A** - 0.79, **BERT-Experiment-7-A** - 0.78 and **GLiNER-Experiment-2** - 0.78 respectively. Observing the sub-columns of the **Dev**-column, one can see the results that drove our decisions in improving the system. The difference between experiment 1 and experiment 2 shows the contribution of the frequency gazetteers for ambiguous entities in comparison with the baseline. This is demonstrated also with experiments 4 and 5. Both have as input the NER annotations of **BERT-Experiment-7-A**, however in Experiment 4 the EL is performed with the baseline system, while in Experiment 5, the improved system using frequency gazetteers is used. Only this change alone raises the μ-F_1 by more than 10 percentage points on the **Dev** and **Test** splits. While the performance of the two ensemble approaches is highly competitive—with differences becoming apparent only beyond two decimal places—**EL Ensemble B** consistently yields a superior μ-F_1 score when evaluated to four decimal digits. It is worth noting that all submitted systems produced with **EL Ensemble A** and **EL Ensemble B** ranked above the BASELINE on the **Test** set. The term BASELINE denotes the official baseline provided by the task organizers for the respective subtask. The system evaluation at the task was done based on μ-F_1; therefore the μ- results are mainly presented here.

Experiment 11 shows the performance of a joint system with a partnering team. The output of the NERD system **Ensemble-Experiment-2-A** & **EL Ensemble A** is united with the output of their **CHASTE** system. This joint result ranked 3rd on the NERD subtask with 0.56 μ-F_1 on the **Test** set.

Table 4
Entity Linking results. Experiment 8 and 11 are in bold because they are ranked 2nd and 3rd respectively according to the official leaderboard.

			Dev				Test			
ID	**EL Model**	**NER Model**	Ma-F_1	μ-P	μ-R	μ-F_1	Ma-F_1	μ-P	μ-R	μ-F_1
1	BASELINE	BASELINE	0.6129	0.6088	0.6537	0.6305	0.39	0.43	0.45	0.44
2	EL Ensemble A	BASELINE	0.6994	0.7192	0.7723	0.7448	0.4842	0.4676	0.5484	0.5793
3	EL Ensemble B	BASELINE	0.6999	0.7200	0.7731	0.7456	0.4898	0.4728	0.5554	0.5867
4	BASELINE	BERT-Experiment-7-A	0.6576	0.6656	0.6815	0.6735	-	-	-	-
5	EL Ensemble A	BERT-Experiment-7-A	0.7379	0.7772	0.7957	0.7864	0.4851	0.5741	0.5830	0.5785
6	EL Ensemble B	BERT-Experiment-7-A	0.7376	0.7772	**0.7957**	0.7864	0.4898	0.5818	**0.5908**	0.5862
7	EL Ensemble A	GLiNER-Experiment-2	0.7409	0.8013	0.7664	0.7835	0.4969	0.6127	0.5812	0.5965
8	**EL Ensemble B**	**GLiNER-Experiment-2**	0.7410	**0.8017**	0.7668	0.7839	**0.5029**	**0.6217**	0.5897	**0.6053**
9	EL Ensemble A	Ensemble-Experiment-2-A	0.7457	0.7949	0.7886	0.7917	0.4854	0.5944	0.5752	0.5847
10	EL Ensemble B	Ensemble-Experiment-2-A	**0.7458**	0.7953	0.7890	**0.7921**	0.4904	0.6025	0.5830	0.5926
11	**EL Ensemble A ∪ CHASTE by ToGS Team**	**Ensemble-Experiment-2-A**	0.74	**0.85**	0.70	0.77	0.45	**0.77**	0.44	0.56

8. Relation Extraction Approaches

8.1. ATLOP

For the relation extraction task, the first method used was Adaptive Thresholding and Localized Context Pooling (ATLOP) [6]. ATLOP builds upon a conventional transformer-based encoder [30, 31] as its backbone architecture. ATLOP depends on entities being supplied externally, meaning predictions can only be generated once the NER model has produced its extractions.

The general procedure begins with an input sequence $d = [x_t]_{t=1}^{l}$ along with a collection of entities $\{e_i\}_{i=1}^{n}$. Entity mention boundaries are marked by inserting a special token "*" immediately before and after each mention. Passing this sequence through a pretrained encoder produces the contextualized token representations:

$$H = [h_1, h_2, \ldots, h_l] = \text{BERT}([x_1, x_2, \ldots, x_l]). \tag{1}$$

Each entity is represented using the embedding of the "*" token preceding its mention, and all mention embeddings belonging to the same entity are aggregated via logsumexp pooling [32]. Given the pooled representations (h_{e_s}, h_{e_o}) for an entity pair (e_s, e_o), the embeddings are first projected into hidden states z through linear layers with tanh activations, after which a bilinear transformation followed by a sigmoid yields the probability of relation r:

$$z_s = \tanh(W_s h_{e_s}), \tag{2}$$

$$z_o = \tanh(W_o h_{e_o}), \tag{3}$$

$$P(r \mid e_s, e_o) = \sigma(z_s^\top W_r z_o + b_r). \tag{4}$$

To overcome the shortcomings of fixed global thresholds in multi-label classification, ATLOP proposes the Adaptive Thresholding (AT) mechanism. Rather than relying on a single static cutoff, AT learns an entity-pair-specific threshold, thereby reducing inference-time misclassifications. This is accomplished by introducing a dedicated "threshold class," referred to as TH, which is trained alongside the genuine relation classes and serves as a learned decision boundary. The primary advantage of AT lies in the fact that the threshold is optimized jointly with the model during training, rather than being tuned *post hoc* by sweeping values in $(0, 1)$ to maximize evaluation metrics. Together with Localized Context Pooling and several additional design choices, these contributions make ATLOP both effective and straightforward to apply.

In an effort to boost performance, a variety of pretrained encoders were evaluated as the underlying backbone, including BERT [31], RoBERTa [33], PubMedBERT [34], BiomedELECTRA [35], BiomedBERT [36], MedCPT [37], and BioLinkBERT [38]. The fine-tuning hyperparameters are reported in Appendix 9. The following datasets are used for training: *Gold*, *Silver*, *Silver-2025* and *Bronze* sets.

We also experimented with parallel encoder ensembles. We created the ensemble by taking 2-3 different encoder models that share the same tokenizer, and we summed their final logits predictions during training before applying the softmax and calculating loss. This proved to be the most effective ATLOP configuration: as reported in Tables 5 and 6, the three-encoder ensemble *(BiomedELECTRA + PubMedBERT + BioLinkBERT)* attains the best scores of any ATLOP variant on both the mention-level (0.442 dev / 0.234 test micro-F_1) and concept-level (0.273 dev / 0.080 test micro-F_1) subtasks, and yields the best macro-F_1 of any system on mention-level RE (0.409 on dev), consistently outperforming every single-encoder backbone.

8.2. LLM Prompting

We additionally experimented with multi-shot prompting of proprietary models from OpenAI's GPT-5 family. Specifically, we report results for two versions of this family, `GPT-5.4` and `GPT-5.5`; the smaller variant `GPT-5-mini` is used only for the label-dictionary generation described in the entity linking approach. For every PubMed abstract, the model receives (i) the title and abstract verbatim and (ii) the list of annotated entity mentions provided by the upstream NER / entity-linking step. For the mention-level task, each entity is described by its *label*, *text span,* and character offsets; for the concept-level task: the offsets are replaced by the concept *URI*. The model is asked to return every relation that holds between two listed entities as a JSON array, where each item is a five-tuple of the form $(s_{\text{text}}, s_{\text{label}}, p, o_{\text{text}}, o_{\text{label}})$ for the mention task and $(s_{\text{uri}}, s_{\text{label}}, p, o_{\text{uri}}, o_{\text{label}})$ for the concept task respectively.

We exploit OpenAI's strict structured-output mode by passing a `response_format` of type `json_schema` that fixes the top-level key (`mention_level_relations` or `concept_level_relations`), restricts *predicate* to the 17 GutBrainIE Task predicates, and restricts *subject_label* / *object_label* to the 13 GutBrainIE Task entity categories. This guarantees a syntactically valid output and removes an entire class of post-processing errors. The constraint that the $(s_{\text{label}}, p, o_{\text{label}})$ triple be drawn from the challenge's allowed combinations cannot be expressed in JSON Schema and is therefore re-stated in natural language inside the system prompt and re-checked offline.

The system prompt is concatenated with K in-context examples sampled deterministically from the train set, restricted to documents that contain at least one relation. Each example shows the exact `(user input, JSON output)` pair the model is expected to imitate. Despite strict decoding, the model occasionally emits triples whose $(s_{\text{label}}, p, o_{\text{label}})$ pattern is not allowed by the schema or whose text span / URI does not appear in the provided entity list. A post-processing step drops such triples, deduplicates exact matches, and produces the submission-format. Temperature is fixed at 0.0. The reasoning effort for reasoning models is set to `medium`.

The mention-level and concept-level system prompts used for this method are shown in Appendix A and Appendix B; the user-message template and the rendered few-shot block are shown in Appendix C.

8.3. LLM Fine-tuning

We experimented with fine-tuning an LLM using LoRA adapters [39] and supervised fine-tuning (SFT). The setup was a system prompt containing the instructions to the model, including limitations on the relations and entities. Additionally, in the user prompt, we provided the title and the document, plus the entities extracted by the NER model. For the concept-level relation extraction subtask, we also provide the concepts annotated by the entity linking model. The loss of the model is calculated only on the completion tokens. The model used was Qwen3.5-4B [7]. The hyperparameters used are shown in Table 10 of Appendix H. Additionally, the LoRA adapters were applied to all linear layers in the attention mechanism and the MLPs.

Every example is a strict three-message conversation (*system, user, assistant*) where the *system* message is the *zero-shot* version of the prompt from Section 8.2, the *user* message is the rendered article-plus-entities block, and the *assistant* message is a JSON string containing the gold relations. We verified during data preparation that no `EXAMPLES:` block is present, so the model is never trained to imitate in-context exemplars. Training documents are merged across the *gold*, *silver*, *silver-2025* and *bronze* GutBrainIE splits, with later splits overriding earlier ones on PMID conflict; documents with empty relation lists are excluded by default. A combined shuffled file mixing both tasks is also produced to enable multi-task SFT.

9. Relation Extraction Results

For relation extraction, we experimented using the gold NER and EL annotations. For the submission, we had to use predictions on the test set from our systems. We mark the predictions used with a suffix, where the suffix mapping is:

- **EXP-2** — `GLiNER-Experiment-2` for NER and `EL Ensemble B` for EL.
- **EXP-4** — `Ensemble-Experiment-2-A` for NER and `EL Ensemble B` for EL.
- **EXP-8** — `BERT-Experiment-7-A` for NER and `EL Ensemble B` for EL.

Mention-level RE. On dev (Table 5), the three model families occupy clearly distinct performance categories. The ATLOP with BERT sits at 0.426 micro-F_1; substituting any of the biomedical pre-trained encoders into the same ATLOP head lifts micro-F_1 into the $0.40 - 0.43$ range, and the three-encoder parallel ensemble is the strongest non-LLM configuration (0.442 micro-F_1, $\mathbf{0.409}$ macro-F_1 the best macro-F_1 of any system on this subtask). The fine-tuned Qwen3.5-4B is the most conservative system in the table: it leads micro-precision at $\mathbf{0.660}$, but its micro-recall (0.315) is essentially at the ATLOP BERT level, which keeps its micro-F_1 (0.427) on par with single-encoder ATLOP variants. Few-shot `GPT-5.5` is the top dev system overall, reaching $\mathbf{0.499}$ micro-F_1 at $K{=}15$ shots, with the highest recall in the table ($\mathbf{0.625}$ micro-recall); $K{=}25$ gives nearly the same micro-F_1 (0.489). The `GPT-5.4` sweep saturates around 0.40–0.42 micro-F_1 regardless of K, confirming that within the in-context-learning family, the model, rather than the shot budget, is the bottleneck. The `GPT-5.4 (25 shots, no entities)` ablation the same prompt with the entity list stripped out collapses to 0.333 micro-F_1 (a 0.082-point drop versus the matching $K{=}25$ run with entities), which quantifies how much the few-shot baseline relies on the upstream NER step, even when shown the full document text.

On test (Table 5), absolute scores drop by roughly one third across the board, as expected, since the test split uses NER + EL predictions rather than gold mentions; a missing or wrong entity makes every triple anchored on it unrecoverable. The relative ordering between families is nonetheless preserved. Few-shot `GPT-5.5` is again the strongest configuration, with three runs tied within 0.001 micro-F_1 at the top: `15 shots, EXP8 entities`, ($\mathbf{0.395}$), `15 shots` (0.395), and `20 shots, EXP2 entities` (0.394, which simultaneously holds the best test macro-F_1 at $\mathbf{0.298}$). Fine-tuned Qwen3.5-4B again leads micro-precision $\mathbf{0.522}$ but is recall-limited (0.245 micro-recall), while the ATLOP family clusters tightly between 0.20 and 0.23 micro-F_1 with the three-encoder ensemble again best (0.234) in the ATLOP family of models.

Concept-level RE. On dev (Table 6), concept-level RE is markedly harder for the ATLOP family: no single-encoder configuration exceeds 0.26 micro-F_1 and even the three-encoder ensemble plateaus at 0.273. Both LLM families clear this gap by a wide margin. The fine-tuned Qwen3.5-4B is the top dev system on macro-F_1 ($\mathbf{0.388}$) and micro-precision ($\mathbf{0.720}$), making it the most conservative concept-level predictor we trained. Few-shot `GPT-5.5` wins on micro-F_1 by a narrow margin: $K{=}20$ reaches $\mathbf{0.472}$ 0.008 points above the Qwen3.5-4B fine-tune and $K{=}15$ yields the highest micro-recall in the table ($\mathbf{0.612}$) at the cost of slightly lower precision. The `GPT-5.4` sweep shows a clean monotone trend up to $K{=}25$ (0.445 micro-F_1) followed by a sharp collapse at $K{=}50$ (0.283); inspection of the prompts

Table 5
Mention-level RE (M-RE, Subtask 6.2.1). Columns: Ma-F_1 = macro-averaged F_1; μ-P, μ-R, μ-F_1 = micro-averaged precision, recall and F_1. Row groups: ATLOP ENCODERS are ATLOP variants that swap in alternative encoders, including two parallel ensembles; FEW-SHOT PROPRIETARY LLM are `GPT-5.4` and `GPT-5.5` predictions made under the prompting protocol of Section 8.2, with the parenthetical giving the number K of in-context shots and the entity-input source: a suffix EXP2/EXP4/EXP8 names one of our internal NER + entity-linking configurations, `no entities` is an ablation that omits the entity list from the prompt entirely, and a missing suffix denotes gold-quality entities on dev or the default NER + EL configuration on test. Best value per column is in bold.

	Dev				Test			
Model	Ma-F_1	μ-P	μ-R	μ-F_1	Ma-F_1	μ-P	μ-R	μ-F_1
ATLOP encoders								
ATLOP, BERT-base	0.343	0.512	0.364	0.426	0.207	0.489	0.128	0.203
ATLOP, RoBERTa-base	0.315	0.484	0.348	0.405	0.196	0.476	0.125	0.198
ATLOP, PubMedBERT(abstract)	0.366	0.483	0.361	0.413	0.209	0.518	0.135	0.214
ATLOP, PubMedBERT(abstract+fulltext)	0.369	0.476	0.361	0.410	0.191	0.475	0.134	0.209
ATLOP, BiomedELECTRA	0.309	0.441	0.368	0.401	0.175	0.466	0.132	0.206
ATLOP, BioLinkBERT-base	0.351	0.452	0.386	0.417	0.202	0.457	0.141	0.215
ATLOP, BioLinkBERT-large	0.338	0.470	0.364	0.410	0.198	0.479	0.135	0.210
ATLOP, MedCPT-Article-Encoder	0.312	0.455	0.375	0.411	0.191	0.449	0.131	0.202
ATLOP, parallel (BiomedELECTRA + BiomedBERT)	0.391	0.454	0.413	0.432	0.202	0.455	0.147	0.223
ATLOP, parallel (BiomedELECTRA + PubMedBERT+ BioLinkBERT)	**0.409**	0.437	0.447	0.442	0.222	0.433	0.161	0.234
Fine-tuned open-weights LLM								
Qwen3.5-4B	0.319	**0.660**	0.315	0.427	0.231	**0.522**	0.245	0.333
Few-shot proprietary LLM								
GPT-5.4 (5 shots)	0.324	0.527	0.329	0.405	–	–	–	–
GPT-5.4 (10 shots)	0.329	0.517	0.338	0.409	–	–	–	–
GPT-5.4 (15 shots)	0.303	0.529	0.323	0.401	–	–	–	–
GPT-5.4 (20 shots)	0.332	0.527	0.355	0.424	–	–	–	–
GPT-5.4 (25 shots)	0.325	0.543	0.336	0.415	–	–	–	–
GPT-5.4 (25 shots, no entities)	0.306	0.474	0.256	0.333	–	–	–	–
GPT-5.4 (50 shots)	0.327	0.511	0.338	0.407	–	–	–	–
GPT-5.5 (15 shots)	0.393	0.415	**0.625**	**0.499**	0.275	0.345	0.425	0.381
GPT-5.5 (15 shots, EXP4 entities)	–	–	–	–	0.277	0.335	0.423	0.374
GPT-5.5 (15 shots, EXP8 entities)	–	–	–	–	0.287	0.352	**0.450**	**0.395**
GPT-5.5 (20 shots, EXP2 entities)	–	–	–	–	**0.298**	0.358	0.439	0.394
GPT-5.5 (20 shots, EXP4 entities)	–	–	–	–	0.263	0.339	0.413	0.372
GPT-5.5 (20 shots, EXP8 entities)	–	–	–	–	0.253	0.322	0.416	0.363
GPT-5.5 (25 shots)	0.396	0.428	0.571	0.489	0.270	0.348	0.445	0.391
GPT-5.5 (25 shots, EXP4 entities)	–	–	–	–	0.263	0.328	0.413	0.366
GPT-5.5 (25 shots, EXP8 entities)	–	–	–	–	0.257	0.327	0.409	0.363

suggests that context-window saturation is to blame past $\sim$25 shots the model starts to degrade in performance. Stripping the entity list out of the `GPT-5.4 (25 shots)` prompt entirely costs 0.127 micro-F_1 ($0.445 \rightarrow 0.318$), a larger drop than the corresponding ablation on the mention task.

The drop from dev to test is steeper than for the mention-level task: the best test configuration reaches 0.245 micro-F_1 versus 0.472 on dev. The concept-level evaluation builds on the evaluation and errors in each of the previous steps - NER, NERD and Mention-level RE. The ranking between families is amplified accordingly: ATLOP encoders bottom out at 0.06–0.08 micro-F_1, the fine-tuned Qwen3.5-4B reaches 0.125, and few-shot `GPT-5.5` dominates with $\mathbf{0.245}$ at $K{=}15$ over EXP4 entities a configuration that simultaneously holds the best test macro-F_1 ($\mathbf{0.181}$), micro-precision ($\mathbf{0.209}$) and micro-recall ($\mathbf{0.296}$). Entity-level performance, such as the quality of NER and EL, influences concept-level extraction to a greater extent than mention-level RE: at $K{=}15$, the default-NER+EL run yields 0.140 micro-F_1 while feeding the *same* prompt the EXP4 entity list lifts micro-F_1 to 0.245 a relative gain of $>70\%$ from the entity linker alone.

Table 6
Concept-level RE (C-RE, Subtask 6.2.2). Column and row-group conventions identical to Table 5. Ma-F_1 = macro-averaged F_1; μ-P, μ-R, μ-F_1 = micro-averaged precision, recall and F_1. Few-shot LLM rows are labelled by the number K of in-context shots (sampled from the gold-quality train split) and by the entity-input source on the test split (EXPx = one of our internal NER + EL configurations, `no entities` = no entity list in the prompt, no suffix = gold entities on dev / default NER + EL on test). Best value per column is in bold.

	Dev				Test			
Model	Ma-F_1	μ-P	μ-R	μ-F_1	Ma-F_1	μ-P	μ-R	μ-F_1
ATLOP encoders								
ATLOP, BERT-base	0.263	0.288	0.231	0.256	0.055	0.129	0.041	0.062
ATLOP, RoBERTa-base	0.212	0.265	0.213	0.236	0.062	0.145	0.047	0.071
ATLOP, PubMedBERT(abstract)	0.251	0.270	0.231	0.249	0.070	0.155	0.049	0.075
ATLOP, PubMedBERT(abstract+fulltext)	0.253	0.274	0.237	0.254	0.063	0.135	0.047	0.070
ATLOP, BiomedELECTRA	0.217	0.255	0.240	0.247	0.051	0.136	0.047	0.070
ATLOP, BioLinkBERT-base	0.240	0.260	0.252	0.256	0.052	0.124	0.046	0.067
ATLOP, BioLinkBERT-large	0.240	0.266	0.235	0.250	0.057	0.120	0.041	0.061
ATLOP, MedCPT-Article-Encoder	0.237	0.261	0.243	0.252	0.047	0.110	0.040	0.058
ATLOP, parallel (BiomedELECTRA + BiomedBERT)	0.270	0.259	0.262	0.261	0.062	0.124	0.049	0.071
ATLOP, parallel (BiomedELECTRA + PubMedBERT+ BioLinkBERT)	0.288	0.255	0.294	0.273	0.064	0.129	0.058	0.080
Fine-tuned open-weights LLM								
Qwen3.5-4B	**0.388**	**0.720**	0.342	0.464	0.102	0.175	0.097	0.125
Few-shot proprietary LLM								
GPT-5.4 (5 shots)	0.295	0.428	0.330	0.373	–	–	–	–
GPT-5.4 (10 shots)	0.327	0.495	0.351	0.411	–	–	–	–
GPT-5.4 (15 shots)	0.333	0.460	0.372	0.411	–	–	–	–
GPT-5.4 (20 shots)	0.321	0.479	0.375	0.420	–	–	–	–
GPT-5.4 (25 shots)	0.341	0.509	0.395	0.445	–	–	–	–
GPT-5.4 (25 shots, no entities)	0.281	0.415	0.258	0.318	–	–	–	–
GPT-5.4 (50 shots)	0.202	0.476	0.201	0.283	–	–	–	–
GPT-5.5 (15 shots)	0.379	0.371	**0.612**	0.462	0.108	0.118	0.173	0.140
GPT-5.5 (15 shots, EXP4 entities)	–	–	–	–	**0.181**	**0.209**	**0.296**	**0.245**
GPT-5.5 (15 shots, EXP8 entities)	–	–	–	–	0.167	0.198	0.274	0.230
GPT-5.5 (20 shots)	0.384	0.393	0.589	**0.472**	0.115	0.125	0.179	0.147
GPT-5.5 (20 shots, EXP2 entities)	–	–	–	–	0.160	0.186	0.272	0.221
GPT-5.5 (20 shots, EXP4 entities)	–	–	–	–	0.149	0.177	0.262	0.211
GPT-5.5 (20 shots, EXP8 entities)	–	–	–	–	0.157	0.183	0.284	0.222
GPT-5.5 (25 shots)	0.373	0.373	0.597	0.460	0.100	0.119	0.171	0.141
GPT-5.5 (25 shots, EXP4 entities)	–	–	–	–	0.152	0.180	0.262	0.214
GPT-5.5 (25 shots, EXP8 entities)	–	–	–	–	0.162	0.186	0.279	0.223

9.1. RE Error Analysis

Qualitative analysis of mention-level relation extraction errors revealed recurrent discrepancy patterns involving (i) predicate directionality ambiguity, (ii) annotation inconsistencies in comparative statements, (iii) ambiguity in containment relations and (iv) competing selection of semantically related spans. In several cases, semantically plausible predictions conflicted with the annotation conventions adopted in the corpus, resulting in paired FP/FN evaluations.

One recurrent source of errors involved inconsistencies in predicate directionality between sentence semantics and annotation structure. For example, in the gold standard the following relation is annotated "MCI patients" $\rightarrow$ "used by" $\rightarrow$ "acupuncture therapy", whereas the model predicted the semantically more rational order "acupuncture" $\rightarrow$ "used by" $\rightarrow$ "MCI patients". In this case, the model also reduced the annotated span from “acupuncture therapy” to “acupuncture”, suggesting an interaction between predicate directionality ambiguity and entity span variation inherited from entity recognition step.

Annotation inconsistencies were also frequent in comparative statements containing multiple ar-

guments. In sentences such as "the abundance of Veillonella was significantly higher in SZ patients than in NCs", the model generated relations linking the bacterial taxon to both participant cohorts, whereas the gold standard encoded only one binary cohort association. Similar patterns were observed for "fungal and bacterial taxa" across multiple abstracts.

Additional discrepancies emerged from ambiguity in containment relations and competing selection of semantically related spans. For example, in the gold standard the annotated "gut microbiota" → "located in" → "ApoE4-KI males" is annotated, whereas the model predicted "gut" → "located in" → "ApoE4-KI males". Both interpretations are semantically plausible because the predicate "located in" may encode multiple types of containment relations, including anatomical containment and microbiome-host association. However, because the evaluation framework evaluates the relation direction, the failure to reproduce the gold-standard relation resulted in a false negative. In other cases, the gold annotations established relations using broad participant-group mentions such as "participants", whereas the model selected more specific subgroup expressions, including "female" and "rural elderly female". Finally, some errors resulted from formatting-sensitive matching discrepancies. In these cases, formatting artifacts and escaped characters caused identical annotated relations to be evaluated as FP or FN, despite the absence of any differences between the gold annotations and the model predictions.

FP/FN analysis suggested that, overall, semantically plausible model predictions indicated promising outputs from data annotation. Nevertheless, there remains room for improvement in subsequent annotation and evaluation stages, including clearer annotation guidelines for relation directionality, more consistent treatment of semantically related mentions, and normalization procedures aimed at reducing formatting-sensitive matching discrepancies.

Qualitative analysis of the concept-level relation extraction errors revealed that most FP and FN discrepancies originated from propagation of mention-level extraction errors during normalization. In the majority of analyzed cases, concept-level mismatches directly inherited semantic ambiguities already present at the mention-level. These findings suggest that concept-level RE performance is strongly dependent on the consistency and semantic stability of the underlying mention-level annotations.

10. Discussion and Conclusion

This thorough work on gut-brain interplay information extraction resulted in 3 2nd places, 1 3rd place and 1 5th place. Here are our top ranked systems on the respective tasks:

- NER (2nd place): GLiNER-Experiment-2
- NERD (2nd place): EL Ensemble B + GLiNER-Experiment-2
- NERD (3rd place): EL Ensemble B + GLiNER-Experiment-2-A ∪ CHASTE by ToGS Team
- M-RE (5th place): GPT-5.5 (15 shots, EXP8 entities)
- C-RE (2nd place): GPT-5.5 (15 shots, EXP4 entities)

10.1. Named Entity Recognition

We experiment with three groups of models, namely **BERT**, **GLiNER**, and **Ensemble**. While on the dev set all **BERT** systems show increasing improvements (from 79.84% to 84.44%), they also experience the largest drop in performance on the test set. However, the biomedical **GLiNER** models demonstrate consistency sometimes achieving even better results on the unseen test data. Due to the **Ensembles** being formed mainly from **BERT** models, they also experience a notable drop on the test set. Finally, our findings suggest a promising research direction for biomedical NER based on the GLiNER architecture without the need for expensive and slow LLMs. They also show increased robustness against common issues like semantic ambiguity, annotation inconsistency, etc. Techniques like weighted training and data augmentation tend to be beneficial. Perhaps the most successful strategy would be to apply the data techniques during the **GLiNER** models fine-tuning, and consequently form ensembles with them. However, we leave this for future work.

10.2. Named Entity Recognition and Disambiguation

The NERD (entity linking) results indicate that the subtask benefits significantly from both **unambiguous and frequency-based gazetteers**. This frequency-based resolution improves P, R and F_1 scores by over 10 percentage points on Dev and Test datasets. For out-of-vocabulary terms, out-of-the-box **PubMedBERT** outperforms fine-tuned **SapBERT**, though an ensemble combining both models yields the highest overall results. Although this improvement is marginal—visible only at the third and fourth decimal places—its consistency across all experiments justifies the ensemble approach. Ultimately, these findings demonstrate the efficacy of hybrid systems, confirming that integrating models via a voting mechanism enhances performance. The efficiency of the ensemble systems is also supported by the results from our joint system with the partnering team from Technical University in Graz.

10.3. Relation Extraction

Across both subtasks and both splits, we see a consistent triangulation of the precision-recall distribution that maps onto the modeling choice rather than the model size. The ATLOP encoder family, including the two parallel ensembles, behaves like a balanced classifier with moderate precision and moderate recall and is the strongest non-generative approach in our study; the three-encoder ensemble retains the best macro-F_1 on mention-level dev (0.409). The fine-tuned Qwen3.5-4B sits at the high-precision corner of the plane on every split and every subtask (e.g. 0.660 dev / 0.522 test micro-precision on mention; 0.720 dev micro-precision on concept), at the cost of a substantially lower recall. Few-shot `GPT-5.5` occupies the opposite, high-recall corner of the plane and delivers the best micro-F_1 on every column of every table where it is evaluated.

While ATLOP requires the entities to be extracted first, we demonstrated that LLM-based methods also achieve significant gains when provided with the extracted entities. Because BERT models generally lack zero-shot capabilities and their size has mostly been capped at one billion parameters, ATLOP models have limited potential for more complex tasks, but are still a good choice for resource-constrained settings. LLMs have greater potential, but as we showed, even top closed-source LLMs benefit from in-context learning, and their performance appears to plateau at around $15 - 25$ shots for this task. Fine-tuned LLMs demonstrate substantial potential as the second-best to closed-source LLMs with in-context learning, as they can scale with model size and training data. For future development, fine-tuning with reinforcement learning has strong potential to reduce the need for additional training data and to produce reasoning-capable LLMs rather than only instruction-tuned ones.

Acknowledgments

This work is partially supported by European Union's Horizon research and innovation programme project HEREDITARY [Grant Agreement No.101137074] and part of this research is supported by the Portuguese national funding through the FCT – Portuguese Foundation for Science and Technology, I.P. as part of the project UID/03213/2025[24], and by the European Union NextGenerationEU, as part of the project UID/PRR/03213/2025[25] – Research Center for Linguistics at NOVA University Lisbon (CLUNL).

Declaration on Generative AI

During the preparation of this paper, the authors used Gemini-Pro in order to: Grammar and spelling check and refine the English and to collect relevant sources of related work. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the publication's content. Various other AI tools are described as part of the methodology in the relevant sections.

[24]https://doi.org/10.54499/UID/03213/2025
[25]https://doi.org/10.54499/UID/PRR/03213/2025

References

[1] M. Martinelli, V. Bonato, G. M. Di Nunzio, G. Faggioli, N. Ferro, O. Irrera, B. Kantz, S. Marchesin, L. Menotti, S. Merlo, R. Michelotto, G. Tussardi, F. Vezzani, P. Waldert, G. Silvello, Overview of GutBrainIE@CLEF 2026: Gut-Brain Interplay Information Extraction, in: E. Sánchez Salido, A. Barrón-Cedeño, A. García Seco de Herrera, S. MacAvaney, J. M. Struß (Eds.), CLEF 2025 Working Notes, 2026.

[2] A. Nentidis, G. Katsimpras, A. Krithara, M. Krallinger, M. Rodríguez-Ortega, E. Rodriguez-López, N. Loukachevitch, I. Rozhkov, E. Tutubalina, D. Dimitriadis, V. Patsiou, 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 2026: The fourteenth BioASQ Challenge on Large-Scale Biomedical Semantic Indexing and Question Answering, volume Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Seventeenth International Conference of the CLEF Association (CLEF 2026) of *Lecture Notes in Computer Science*, Springer, 2026.

[3] U. Zaratiana, N. Tomeh, P. Holat, T. Charnois, GLiNER: Generalist model for named entity recognition using bidirectional transformer, in: K. Duh, H. Gomez, S. Bethard (Eds.), Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), Association for Computational Linguistics, Mexico City, Mexico, 2024, pp. 5364–5376. URL: https://aclanthology.org/2024.naacl-long.300/. doi:10.18653/v1/2024.naacl-long.300.

[4] Y. Gu, R. Tinn, H. Cheng, M. Lucas, N. Usuyama, X. Liu, T. Naumann, J. Gao, H. Poon, Domain-specific language model pretraining for biomedical natural language processing, ACM Trans. Comput. Healthcare 3 (2021). URL: https://doi.org/10.1145/3458754. doi:10.1145/3458754.

[5] F. Liu, E. Shareghi, Z. Meng, M. Basaldella, N. Collier, Self-alignment pretraining for biomedical entity representations, in: K. Toutanova, A. Rumshisky, L. Zettlemoyer, D. Hakkani-Tur, I. Beltagy, S. Bethard, R. Cotterell, T. Chakraborty, Y. Zhou (Eds.), Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Online, 2021, pp. 4228–4238. URL: https://aclanthology.org/2021.naacl-main.334/. doi:10.18653/v1/2021.naacl-main.334.

[6] W. Zhou, K. Huang, T. Ma, J. Huang, Document-level relation extraction with adaptive thresholding and localized context pooling, 2020. URL: https://arxiv.org/abs/2010.11304. arXiv:2010.11304.

[7] A. Yang, A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Gao, C. Huang, C. Lv, C. Zheng, D. Liu, F. Zhou, F. Huang, F. Hu, H. Ge, H. Wei, H. Lin, J. Tang, J. Yang, J. Tu, J. Zhang, J. Yang, J. Yang, J. Zhou, J. Zhou, J. Lin, K. Dang, K. Bao, K. Yang, L. Yu, L. Deng, M. Li, M. Xue, M. Li, P. Zhang, P. Wang, Q. Zhu, R. Men, R. Gao, S. Liu, S. Luo, T. Li, T. Tang, W. Yin, X. Ren, X. Wang, X. Zhang, X. Ren, Y. Fan, Y. Su, Y. Zhang, Y. Zhang, Y. Wan, Y. Liu, Z. Wang, Z. Cui, Z. Zhang, Z. Zhou, Z. Qiu, Qwen3 technical report, 2025. URL: https://arxiv.org/abs/2505.09388. arXiv:2505.09388.

[8] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, J. Kang, Biobert: a pre-trained biomedical language representation model for biomedical text mining, CoRR abs/1901.08746 (2019). URL: http://arxiv.org/abs/1901.08746. arXiv:1901.08746.

[9] A. Yazdani, I. Stepanov, D. Teodoro, Gliner-biomed: A suite of efficient models for open biomedical named entity recognition, Bioinformatics (2026). doi:10.1093/bioinformatics/btag322.

[10] R. Mehta, Enhancing biomedical named entity recognition using gliner-biomed with targeted dictionary-based post-processing for bioasq 2025 task 6, 2025. doi:10.48550/arxiv.2510.08588. arXiv:2510.08588.

[11] H. Huang, C. Yuan, Q. Liu, Y. Cao, Document-level relation extraction via separate relation representation and logical reasoning, ACM Transactions on Information Systems (TOIS) 42 (2023) 1–24. doi:10.1145/3597610.

[12] X. Xu, C. Li, H. Xiang, L. Qi, X. Zhang, W. Dou, Attention based document-level relation extraction with none class ranking loss, in: Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, 2024, pp. 6569–6577. doi:10.24963/ijcai.2024/726.

[13] H. Wu, G. Zhou, Y. Xia, H. Liu, T. Zhang, Self-distillation framework for document-level relation extraction in low-resource environments, PeerJ Computer Science 10 (2024) e1930. doi:10.7717/peerj-cs.1930.
[14] L. Ramshaw, M. Marcus, Text chunking using transformation-based learning, in: Third Workshop on Very Large Corpora, 1995. URL: https://aclanthology.org/W95-0107/.
[15] J. Han, Y. Liu, Gutuzh at clef2025 bioasq task 6: a method of sota performance with the best results at gutbrainie ner subtask 1, in: CLEF 2025 Working Notes – Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2025), volume 4038 of *CEUR Workshop Proceedings*, ceur-ws.org, 2025. URL: https://ceur-ws.org/Vol-4038/paper_20.pdf.
[16] L. R. Andersen, M. H. Dolmer, M. I. Gardshodn, J. M. Rodriguez, D. Dell'Aglio, Trusting gut instincts: Transformer-based extraction of structured data from gut-brain axis publications, in: CLEF 2025 Working Notes – Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2025), volume 4038 of *CEUR Workshop Proceedings*, ceur-ws.org, 2025. URL: https://ceur-ws.org/Vol-4038/paper_6.pdf.
[17] A. Datseris, M. Kuzmanov, I. Nikolova-Koleva, D. Taskov, S. Boytcheva, Graphwise @ clef-2025 gutbrainie: Towards automated discovery of gut-brain interactions - deep learning for ner and relation extraction from pubmed abstracts, in: CLEF (Working Notes), 2025, pp. 226–242. URL: https://ceur-ws.org/Vol-4038/paper_14.pdf.
[18] 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.
[19] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, CoRR abs/1810.04805 (2018). URL: http://arxiv.org/abs/1810.04805. arXiv:1810.04805.
[20] Y. Gu, R. Tinn, H. Cheng, M. Lucas, N. Usuyama, X. Liu, T. Naumann, J. Gao, H. Poon, Domain-specific language model pretraining for biomedical natural language processing, CoRR abs/2007.15779 (2020). URL: https://arxiv.org/abs/2007.15779. arXiv:2007.15779.
[21] S. Alrowili, K. Vijay-Shanker, BioM-transformers: Building large biomedical language models with BERT, ALBERT and ELECTRA, in: D. Demner-Fushman, K. B. Cohen, S. Ananiadou, J. Tsujii (Eds.), Proceedings of the 20th Workshop on Biomedical Language Processing, Association for Computational Linguistics, Online, 2021, pp. 221–227. URL: https://aclanthology.org/2021.bionlp-1.24/. doi:10.18653/v1/2021.bionlp-1.24.
[22] K. Clark, M.-T. Luong, Q. V. Le, C. D. Manning, Electra: Pre-training text encoders as discriminators rather than generators, 2020. URL: https://arxiv.org/abs/2003.10555. arXiv:2003.10555.
[23] J. Omura, On the viterbi decoding algorithm, IEEE Transactions on Information Theory 15 (1969) 177–179. doi:10.1109/TIT.1969.1054239.
[24] C. B. Vennerød, A. Kjærran, E. S. Bugge, Long short-term memory rnn, 2021. URL: https://arxiv.org/abs/2105.06756. arXiv:2105.06756.
[25] D. Hendrycks, K. Gimpel, Bridging nonlinearities and stochastic regularizers with gaussian error linear units, CoRR abs/1606.08415 (2016). URL: http://arxiv.org/abs/1606.08415. arXiv:1606.08415.
[26] P. He, X. Liu, J. Gao, W. Chen, Deberta: Decoding-enhanced bert with disentangled attention, 2021. URL: https://arxiv.org/abs/2006.03654. arXiv:2006.03654.
[27] A. Yazdani, I. Stepanov, D. Teodoro, Gliner-biomed: A suite of efficient models for open biomedical named entity recognition, 2025. URL: https://arxiv.org/abs/2504.00676. arXiv:2504.00676.
[28] S. Bogdanov, A. Constantin, T. Bernard, B. Crabbé, E. Bernard, Nuner: Entity recognition encoder pre-training via llm-annotated data, 2024. arXiv:2402.15343.
[29] National Library of Medicine, UMLS Knowledge Sources, [dataset on the Internet], 2024. URL: https://www.nlm.nih.gov/research/umls/licensedcontent/umlsknowledgesources.html, release 2024AA. Bethesda (MD): National Library of Medicine (US).
[30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, 2023. URL: https://arxiv.org/abs/1706.03762. arXiv:1706.03762.
[31] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers

for language understanding, 2019. URL: https://arxiv.org/abs/1810.04805. arXiv:1810.04805.
[32] R. Jia, C. Wong, H. Poon, Document-level n-ary relation extraction with multiscale representation learning, in: J. Burstein, C. Doran, T. Solorio (Eds.), Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 3693–3704. URL: https://aclanthology.org/N19-1370/. doi:10.18653/v1/N19-1370.
[33] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized bert pretraining approach, 2019. URL: https://arxiv.org/abs/1907.11692. arXiv:1907.11692.
[34] Y. Gu, R. Tinn, H. Cheng, M. Lucas, N. Usuyama, X. Liu, T. Naumann, J. Gao, H. Poon, Domain-specific language model pretraining for biomedical natural language processing, 2020. arXiv:arXiv:2007.15779.
[35] K. r. Kanakarajan, B. Kundumani, M. Sankarasubbu, BioELECTRA:pretrained biomedical text encoder using discriminators, in: D. Demner-Fushman, K. B. Cohen, S. Ananiadou, J. Tsujii (Eds.), Proceedings of the 20th Workshop on Biomedical Language Processing, Association for Computational Linguistics, Online, 2021, pp. 143–154. URL: https://aclanthology.org/2021.bionlp-1.16/. doi:10.18653/v1/2021.bionlp-1.16.
[36] S. Chakraborty, E. Bisong, S. Bhatt, T. Wagner, R. Elliott, F. Mosconi, BioMedBERT: A pre-trained biomedical language model for QA and IR, in: D. Scott, N. Bel, C. Zong (Eds.), Proceedings of the 28th International Conference on Computational Linguistics, International Committee on Computational Linguistics, Barcelona, Spain (Online), 2020, pp. 669–679. URL: https://aclanthology.org/2020.coling-main.59/. doi:10.18653/v1/2020.coling-main.59.
[37] Q. Jin, W. Kim, Q. Chen, D. C. Comeau, L. Yeganova, W. J. Wilbur, Z. Lu, Medcpt: Contrastive pre-trained transformers with large-scale pubmed search logs for zero-shot biomedical information retrieval, Bioinformatics 39 (2023). URL: http://dx.doi.org/10.1093/bioinformatics/btad651. doi:10.1093/bioinformatics/btad651.
[38] M. Yasunaga, J. Leskovec, P. Liang, Linkbert: Pretraining language models with document links, 2022. URL: https://arxiv.org/abs/2203.15827. arXiv:2203.15827.
[39] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, Lora: Low-rank adaptation of large language models, 2021. URL: https://arxiv.org/abs/2106.09685. arXiv:2106.09685.

A. Mention-level System Prompt

The mention-level system prompt is built by interpolating the challenge's predicate vocabulary, entity-label vocabulary and the table of allowed $(s_{\text{label}}, p, o_{\text{label}})$ patterns into the template below.

```
You extract mention-level biomedical relations.

You will receive:
  1. A PubMed article (title + abstract).
  2. A list of annotated entity mentions with their label, text span, and character offsets.

Your job: output every relation that holds between two listed entity mentions, as a JSON object
with a single key ‘mention_level_relations‘ whose value is an array of objects shaped like:
  {"subject_text_span": str, "subject_label": str, "predicate": str,
   "object_text_span": str, "object_label": str}

Hard rules:
- Both subject and object MUST be entity mentions from the provided list (copy ‘text_span‘ and
  ‘label‘ verbatim, keeping the same letter case).
- ‘predicate‘ MUST be one of: administered, affect, change abundance, change effect,
  change expression, compared to, impact, influence, interact, is a, is linked to, located in,
  part of, produced by, strike, target, used by.
- ‘subject_label‘ / ‘object_label‘ MUST be one of: anatomical location, animal, bacteria,
```

```
  biomedical technique, chemical, dietary supplement, DDF, drug, food, gene, human,
  microbiome, statistical technique.
- Only emit a triple if the (subject_label, predicate, object_label) pattern is allowed by the
  schema below. Direction matters.
- Emit each distinct triple once. Do not invent entities. If no relation applies, return an
  empty list.

Allowed (subject_label, predicate, object_label) patterns:
- administered: (chemical -> human), (chemical -> animal), (drug -> human), (drug -> animal),
  (food -> human), (food -> animal), (dietary supplement -> human),
  (dietary supplement -> animal)
- affect: (DDF -> DDF)
- change abundance: (DDF -> bacteria), (DDF -> microbiome)
- change effect: (drug -> DDF)
- change expression: (bacteria -> gene), (chemical -> gene), (food -> gene),
  (microbiome -> gene)
- compared to: (microbiome -> microbiome)
- impact: (chemical -> microbiome), (microbiome -> microbiome),
  (chemical -> bacteria), (dietary supplement -> bacteria),
  (dietary supplement -> microbiome), (drug -> bacteria), (drug -> microbiome),
  (food -> bacteria), (food -> microbiome)
- influence: (bacteria -> DDF), (chemical -> DDF), (dietary supplement -> DDF),
  (drug -> DDF), (food -> DDF)
- interact: (bacteria -> bacteria), (bacteria -> chemical), (bacteria -> drug),
  (chemical -> bacteria), (chemical -> chemical), (chemical -> drug),
  (drug -> bacteria), (drug -> chemical), (drug -> drug), (DDF -> chemical),
  (chemical -> DDF)
- is a: (DDF -> DDF)
- is linked to: (microbiome -> DDF)
- located in: (anatomical location -> human), (anatomical location -> animal),
  (bacteria -> human), (bacteria -> animal), (chemical -> human), (chemical -> animal),
  (microbiome -> human), (microbiome -> animal)
- part of: (bacteria -> microbiome), (chemical -> microbiome),
  (microbiome -> bacteria), (microbiome -> chemical)
- produced by: (chemical -> chemical)
- strike: (DDF -> anatomical location)
- target: (DDF -> human), (DDF -> animal)
- used by: (biomedical technique -> human), (biomedical technique -> animal),
  (biomedical technique -> microbiome)

Return STRICT JSON only (no prose, no markdown).
```

B. Concept-level System Prompt

The concept-level system prompt differs from the mention-level one only in the output schema (URIs replace text spans) and in the merge rule that consolidates mentions sharing a URI; the rule block and the table of allowed triples are identical.

```
You extract concept-level biomedical relations for the GutBrainIE 2026 challenge.

You will receive:
  1. A PubMed article (title + abstract).
  2. A list of annotated entity mentions, each with its label, text span, and concept URI.

Your job: output every relation that holds between two concept URIs drawn from the list, as a
JSON object with a single key ‘concept_level_relations‘ whose value is an array of objects shaped
like:
  {"subject_uri": str, "subject_label": str, "predicate": str,
   "object_uri": str, "object_label": str}

Hard rules:
- Both subject and object URIs MUST appear in the provided entity list (copy ‘uri‘ and ‘label‘
```

```
  verbatim, keeping the same letter case).
- Merge mentions that share the same URI: emit each distinct concept-level triple at most once.
- ‘predicate‘ MUST be one of: <17 challenge predicates, same list as Appendix A>.
- ‘subject_label‘ / ‘object_label‘ MUST be one of: <13 entity labels, same list as Appendix A>.
- Only emit a triple if the (subject_label, predicate, object_label) pattern is allowed by the
  schema below. Direction matters.
- Do not invent concepts. If no relation applies, return an empty list.

Allowed (subject_label, predicate, object_label) patterns:
  <identical to Appendix A>

Return STRICT JSON only (no prose, no markdown).
```

C. Shared User Message and Few-Shot Block

The user message is identical between the few-shot baseline and the SFT training data; the only difference is whether the system prompt ends in an `EXAMPLES:` block ($K>0$) or not (SFT, $K=0$). The template below is the mention-level variant; the concept-level variant inlines a `uri` field instead of `location/start/end`.

```
TITLE: <article title>

ABSTRACT: <article abstract>

ENTITIES:
[0] label='<label>', text_span='<surface>', location='<title|abstract>',
    start=<int>, end=<int>
[1] label='<label>', text_span='<surface>', location='<title|abstract>',
    start=<int>, end=<int>
...

Return the mention-level relations as JSON.
```

The few-shot block, appended to the system prompt at inference time when $K>0$, has the shape:

```
EXAMPLES:
(Each example shows the INPUT the user will send and the exact JSON OUTPUT
 you should produce.)

--- EXAMPLE 1 ---
INPUT:
<rendered user message for shot 1>

OUTPUT:
{"mention_level_relations": [ {"subject_text_span": "...", ... }, ... ]}

--- EXAMPLE 2 ---
...

--- END OF EXAMPLES ---
```

D. SFT Chat-Format Target

Each training example for the SFT baseline is stored as a single JSON line of the form below. The `system` field is the zero-shot variant of Appendix A or B; loss is computed only over the `assistant` response.

```
{
  "pmid": "<PMID>",
```

```
  "task": "mention" | "concept",
  "messages": [
    {"role": "system",
     "content": "<zero-shot system prompt from Appendix A or B>"},
    {"role": "user",
     "content": "TITLE: ...\n\nABSTRACT: ...\n\nENTITIES:\n[0] ...\n\nReturn the mention-level
         relations as JSON."},
    {"role": "assistant",
     "content": "{\"mention_level_relations\": [ {\"subject_text_span\": \"alpha-SMA\", \"
         subject_label\": \"chemical\", \"predicate\": \"located in\", \"object_text_span\": \"
         colon\", \"object_label\": \"anatomical location\"}, ... ]}"}
  ]
}
```

E. NER Experiments Hyperparameters

Table 7 shows the hyperparameters used for the **BERT** group. Similarly, Table 8 shows the hyperparameters for the **GLiNER** group.

Table 7
Hyperparameter settings used for the BERT experiments. LR_{bb}, LR_{dense}, and LR_{crf} denote the learning rates of the backbone, dense layer, and CRF layer, respectively. WD denotes weight decay, and Pat. denotes early-stopping patience. Seed-A refers to the random seed in the corresponding experiment variant A, Seed-B in B. In **BERT-Exp. 8** the learning rates are kept the same for both backbones and both dense layers.

Experiment	LR_{bb}	LR_{dense}	LR_{crf}	WD	Sched.	Batch	Epochs / Pat.	Seed-A	Seed-B
BERT-Exp. 1	2×10^{-5}	2×10^{-5}	2×10^{-5}	0.01	Linear	16	20/3	11	17
BERT-Exp. 2	2×10^{-5}	2×10^{-4}	2×10^{-3}	0.01	Linear	16	25/3	17	123
BERT-Exp. 3	2×10^{-5}	2×10^{-4}	2×10^{-3}	0.01	Linear	16	20/3	42	17
BERT-Exp. 4	2×10^{-5}	2×10^{-4}	2×10^{-3}	0.01	Linear	16	25/3	17	11
BERT-Exp. 5	2×10^{-5}	2×10^{-4}	2×10^{-3}	0.01	Linear	8	20/3	17	42
BERT-Exp. 6	2×10^{-5}	2×10^{-4}	2×10^{-3}	0.01	Linear	16	20/3	17	123
BERT-Exp. 7	2×10^{-5}	2×10^{-4}	2×10^{-3}	0.01	Linear	8	20/3	17	17
BERT-Exp. 8	2×10^{-5}	2×10^{-4}	2×10^{-3}	0.01	Linear	16	20/3	17	11

Table 8
Hyperparameter settings used for the GLiNER experiments. No random seed is reported. The confidence thresholds for predictions are determined empirically by evaluating the models on the dev set with different values. Only the best value for each model is reported.

Hyperparameter	Value / Range
Learning rate	5×10^{-6}
Weight decay	0.01
LR scheduler	Cosine
Warmup ratio	0.1
Train/eval batch size	4–8
Training steps	3600–5400
Prediction threshold	0.6–0.7

F. Data Statistics

Figure 4 shows the frequency of the NER labels in the dataset. The most common category is DDF and the least are `food`, `gene`, `statistical technique`. The dataset is formed by a combination of the Gold, Silver, Silver 2025, Broze and Dev sets.

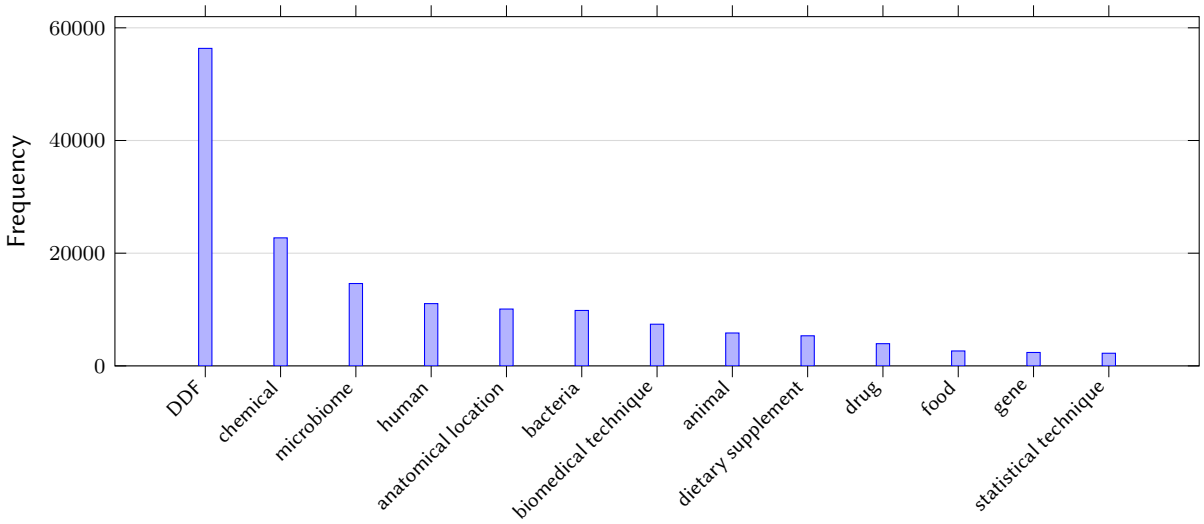

Figure 4: Frequency distribution of entity labels in the dataset (Gold, Silver, Silver 2025, Broze and Dev subsets).

G. Orchestration of resources for EL

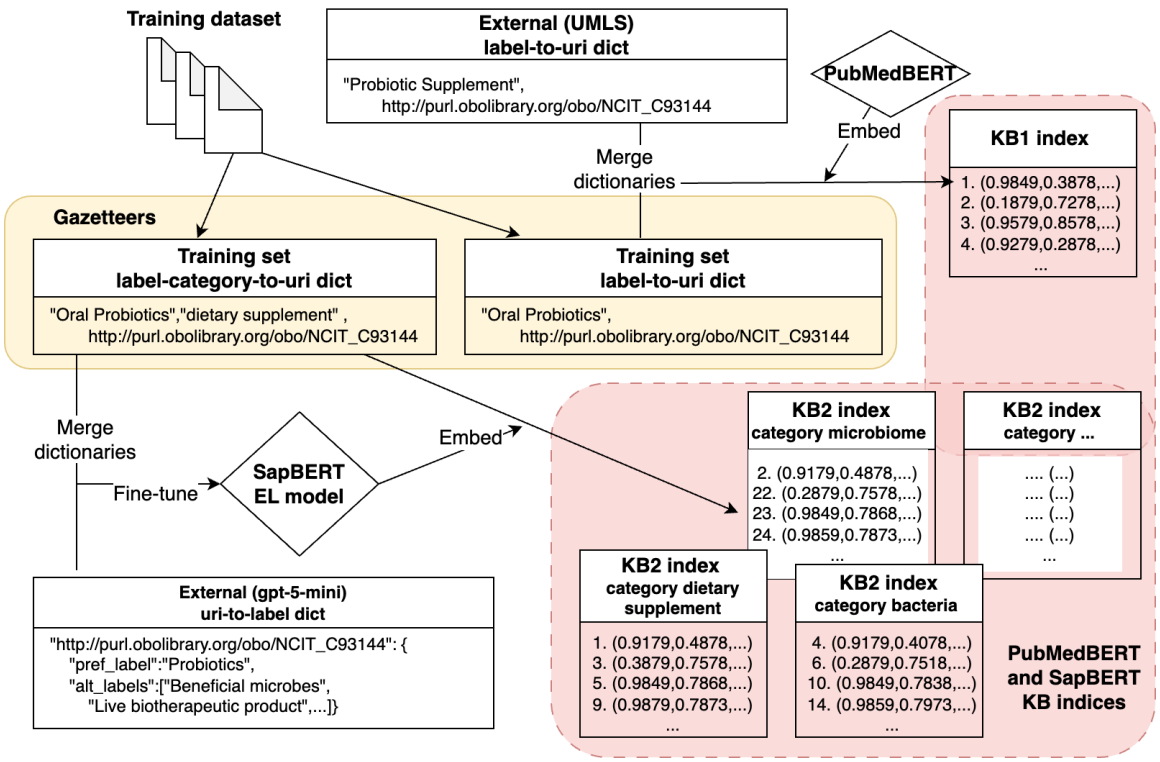

Figure 5: Orchestration of the resources used in the top performing EL system.

H. RE Experiments Hyperparameters

Table 9 shows the hyperparameters for the **ATLOP** models.

Table 10 shows the hyperparameters used for the **Qwen3.5-4B** model training.

Table 9
Hyperparameters used for ATLOP fine-tuning.

Hyperparameter	Value
Batch size	4
Number of epochs	20
Encoder learning rate	5×10^{-5}
Classifier learning rate	1×10^{-4}
Warmup ratio	6%
Max gradient norm	1
Optimizer	AdamW

Table 10
LLM LoRA fine-tuning hyperparameters.

Hyperparameter	Value
Maximum sequence length	16,384
LoRA rank (r)	16
LoRA scaling (α)	16
Per-device batch size	4
Gradient accumulation steps	4
Effective batch size	16
Number of epochs	4
Learning rate	2×10^{-4}
Warmup steps	50

I. Hardware Setup for NER

All our models are domain-specific and have below 500M parameters. Such LMs have both pros and cons. However, they allow us to conduct extensive experimental tests and include more solutions. Our hardware setup includes a single NVIDIA RTX 6000 Ada Generation GPU with 48GB of RAM for training. For inference, a remote AMD Ryzen Threadripper PRO 7975WX CPU with 32-Cores and 503 GB of RAM is used. As inference in general requires much less compute power, it runs smoothly on a local 13th Gen Intel(R) Core(TM) i7-13700H with 32 GB of RAM, too.