Named Entity Recognition with GLiNER and Relation Extraction with LLMs

Notebook for the BioASQ Task GutBrainIE Lab at CLEF 2025

Samuel Piron[1], Giorgio Maria Di Nunzio[1]

[1]Department of Information Engineering, University of Padua, Padova, Italy

Abstract

Biomedical Information Extraction from Natural Language Processing (NLP) is one of the newest challenges driving innovation in the biomedical scientific field. In this work, we present our implementation pipeline for the GutBrain shared task covering both Named Entity Recognition and Relation Extraction. For Subtask 6.1 (NER), we fine-tuned the GLiNER framework on expert-annotated GutBrain datasets, achieving robust entity recognition between the predefined categories. For the RE Subtasks (6.2.1-6.2.3), we injected entity markers into text and employed fine-tuned BiomedBERT and pubmed-bert classifiers to predict relations between entities. By exploring Precision-oriented, Recall-oriented, and balanced configurations, we identified the best setups for maximizing Precision, Recall, and F1 for each task. Finally, we show our results with scatter plots and discuss the trade-off each run offers.

Keywords

Information Extraction, Named Entity Recognition, GLiNER, Relation Extraction, Large Language Models

1. Introduction

In the biomedical field, large volumes of textual data are generated daily, such as electronic health records and biomedical literature. Extracting and structuring this information in an efficient way is crucial for improving healthcare quality, supporting clinical decision-making, and advancing medical research [1]. Natural Language Processing (NLP) techniques have therefore become fundamental tools in medical text mining and Information Extraction (IE).

A key subtask of information extraction is Named Entity Recognition (NER), which involves identifying and categorizing spans of text into predefined categories such as diseases, treatments, and anatomical entities [2].

Based on NER, Relation Extraction (RE) identifies and extracts relationships between named entities from the underlying content [3]. RE is crucial for facilitating the extraction of information from large datasets, particularly when the data is unstructured. It supports many downstream applications, such as transforming unstructured corpora into knowledge graphs, question answering, and automated document processing [4, 5, 6, 7]. Figure 1 illustrates an example of NER and RE annotation in biomedical text.

The CLEF 2025 conference is the 16th edition of the Conference and Labs of the Evaluation Forum (CLEF), continuing the popular CLEF campaigns that have been running since 2000 and contributing to the systematic evaluation of information access systems through experimentation with shared tasks[1]. In particular, BioASQ 2025 Lab Task 6, known as GutBrainIE, promotes the development of Information Extraction systems by extracting named entities and the relations between them[2] [8].

The challenge is divided into 4 Subtasks:

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

samuel.piron@studenti.unipd.it (S. Piron); giorgiomaria.dinunzio@unipd.it (G. Di Nunzio)
https://www.dei.unipd.it/~dinunzio/ (G. Di Nunzio)
0009-0006-5558-4295 (S. Piron); 0000-0001-9709-6392 (G. Di Nunzio)

[1] https://clef2025.clef-initiative.eu/

[2] https://hereditary.dei.unipd.it/challenges/gutbrainie/2025/

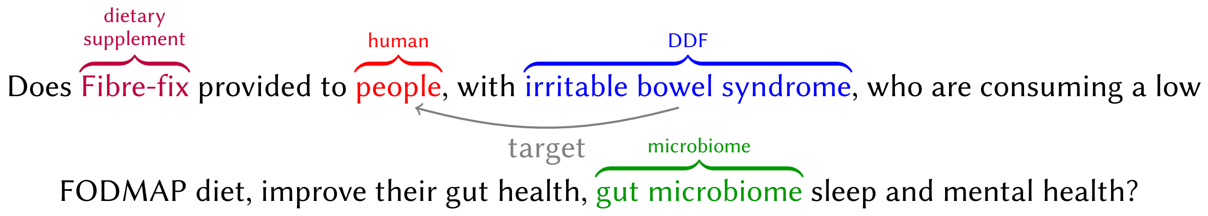
dietary supplement
human
DDF
Does Fibre-fix provided to people, with irritable bowel syndrome, who are consuming a low
target
microbiome
FODMAP diet, improve their gut health, gut microbiome sleep and mental health?

Figure 1: Example of NER and RE in a general biomedical sentence.

Table 1
Example of NER on the GutBrainIE challenge.

Mention	Label
Fibre-fix	Dietary Supplement
People	Human
Irritable Bowel Syndrome	DDF
Gut Microbiome	Microbiome

Table 2
Example of Binary Tag-Based RE on the GutBrainIE challenge.

Subject Label	Object Label
DDF	Human

Table 3
Example of Ternary Tag-Based RE on the GutBrainIE challenge.

Subject Label	Predicate	Object Label
DDF	Target	Human

- Subtask 6.1, the participants are provided with PubMed abstracts about the gut-brain axis focusing on the Parkinson's disease and mental health. Their task is to classify specific entity mentions into one of the 13 predefined categories: anatomical location, animal, biomedical technique, bacteria, chemical, dietary supplement, disease disorder or finding (DDF), drug, food, gene, human, microbiome, and statistical technique. An example is shown in Table 1.
- The next challenges concerns RE over the entities identified by the NER systems:
  - Subtask 6.2.1 is the Binary Tag-Based RE, where participants are asked to identify which entities are in relation within a document. An example is shown in Table 2.
  - Subtask 6.2.2 is the Ternary Tag-Based RE: participants are required to identify the actual entities involved in a relation and predict the type of relation. An example is shown in Table 3.
  - Finally, Subtask 6.2.3, the Ternary Mention-Based RE, where participants are required to identify the actual entities involved in a relation and predict the type of relation. An example is shown in Table 4.

For more detailed information about the GutBrainIE task and the proposed subtasks, please refer to the overview paper [9].

2. Related Work

Early approaches to NER and RE relied on hand-crafted rules, which were later replaced by probabilistic models such as Hidden Markov Models (HMMs) [10] and Conditional Random Fields (CRFs) [11].

Table 4
Example of Ternary Mention-Based RE on the GutBrainIE challenge.

Subject Text Span	Subject Label	Predicate	Object Text Span	Object Label
DDF	Irritable Bowel Syndrome	Target	People	Human

Mikolov et al. [12] introduced distributed word representations, marking one of the first significant advances in capturing word similarity, which remains foundational in modern NLP. Lample et al. [13] proposed a BiLSTM-CRF architecture for NER, where a Bidirectional Long Short-Term Memory network is inserted between the input words and the CRF output layer.

RE first appeared prominently in SemEval-2010 Task 8 [14], which focused on the "Relation Classification Subtask" and assigned a single label to a marked entity pair. The deep learning era began with convolutional neural networks (CNNs), which enabled mapping entire sentences to relation labels without manual feature engineering [15]. Subsequently, the advent of pre-trained language models (PLMs) marked a major advancement in RE, as highlighted by Li et al. [16]. These fine-tuned models, such as BERT and RoBERTa, trained on diverse datasets, have become standard for many modern NLP tasks.

3. Methodology

NER models are widely used in data mining, textual analysis, and text processing. However, they often lack flexibility, and training them can be a challenging task. To address these limitations, Zaratiana et al. [17] proposed GLiNER, a recent and effective alternative to traditional NER models, which are typically restricted to predefined entity types and rely on expensive Large Language Models (LLMs). GLiNER is a NER model capable of recognizing a wide range of entity types using a Bidirectional Transformer architecture.

The model employs a Bidirectional Language Model (BiLM) and takes as input a set of entity type prompts and a sentence or text, with each entity separated by a learned token [ENT]. The BiLM outputs representations for each token. Entity embeddings are passed into a FeedForward Network, where input word representations are passed into a span representation layer to compute embeddings for each span. Finally, it computes a matching score between entity representations and span representations (using dot product and sigmoid activation) [17]. Figure 2 shows the overall architecture.

In our implementation to perform NER on the GutBrain datasets, we used NuNER Zero, a zero-shot NER model[3]. NuNER Zero is based on the GLiNER architecture and expects input as a concatenation of entity types and the target text. It was trained on the NuNER-v2.0 dataset[4], which combines annotated subsets of the Pile[5] and C4[6] corpora. Annotations were generated using large language models following the NuNER procedure, which employs GPT-3.5-turbo to label entity mentions in a large-scale English corpus (C4) [18] with semantically meaningful concepts. The LLM was prompted to extract as many relevant entities as possible from each sentence and assign them to one of approximately 200k unique concepts (e.g., "wellness"). This annotation process resulted in over 4.3 million labeled entities, providing high conceptual diversity but also showing class imbalance and ambiguities, which were addressed through filtering and training procedures [19].

Our implementation pipeline follows these steps:

- **Data analysis**: we begin by examining the composition of the GutBrain corpus (titles, abstracts, and entity annotations), dropping any annotations whose location is not "title" or "abstract".
- **Data loading**: load the JSON files containing metadata and character-level entity labels.

[3] https://huggingface.co/numind/NuNER_Zero
[4] https://huggingface.co/numind/NuNER-v2.0
[5] https://huggingface.co/datasets/EleutherAI/pile
[6] https://huggingface.co/datasets/allenai/c4

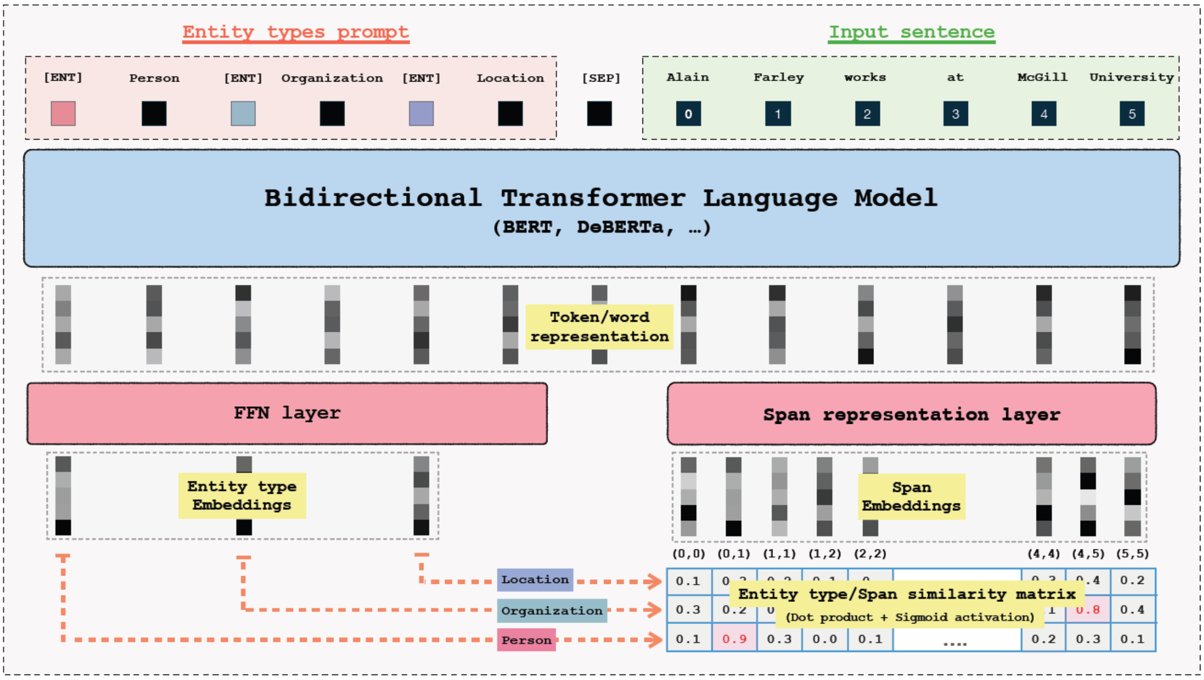

Figure 2: GLiNER Model Processing, from the GLiNER paper (https://arxiv.org/abs/2311.08526).
For example, the span representation of (0,1), corresponding to "Alain Farley", has a high matching score with the entity embeddings of "Person".

- **Tokenizer initialization**: initialize the `microsoft/deberta-v3-large` tokenizer[7], which splits the text into tokens and provides offset mappings for model input.
- **Preprocessing**: combine the title and abstract for each document; convert character-level entity spans into token-level spans using the tokenizer's offset mappings; store the tokenized text and corresponding entities.
- **Hyperparameter configuration**: define the number of training steps based on dataset size, along with the maximum number of epochs, batch size, and maximum token length.
- **Model setup**: initialize the token classification model `numind/NuNER_Zero` based on GLiNER with pre-trained weights, and configure sampling parameters to manage training complexity.
- **Training loop**: iterate over batches, perform forward passes, compute loss, and apply backpropagation. Update model weights using the AdamW optimizer[8].
- **Evaluation**: periodically evaluate model performance on `dev.json`, reporting Micro and Macro Precision, Recall, and F1 scores.

To predict NER entities with our trained models, we structured our implementation pipeline in this way: for each article, we ran the model separately on its title and abstract, requesting predictions for all thirteen entity types (e.g., animal, anatomical location, DDF). The raw output consisted of a list of spans with start and end offsets, labels, and confidence scores. We then merged any adjacent spans with the same label and converted the character-level offsets into the required format. Finally, we saved the predictions in a JSON file, including the (start_idx) and (end_idx) fields.

For the other RE Subtasks of the GutBrainIE challenge, we employed two different pre-trained models: `NeuML/pubmedbert-base-embeddings`[9], a fine-tuned model based on sentence-transformers trained on a dataset of randomly sampled PubMed[10] title–abstract pairs and similar titles; and `microsoft/BiomedNLP-BiomedBERT-base-uncased-abstract-fulltext`[11], a biomedical

[7] https://huggingface.co/microsoft/deberta-v3-large
[8] https://pytorch.org/docs/stable/generated/torch.optim.AdamW.html
[9] https://huggingface.co/NeuML/pubmedbert-base-embeddings
[10] https://pubmed.ncbi.nlm.nih.gov/
[11] https://huggingface.co/microsoft/BiomedCLIP-PubMedBERT_256-vit_base_patch16_224

model pre-trained from scratch using PubMed abstracts and full-text articles from PubMed Central[12] [20]. Our pipeline for these Subtasks is as follows:

- **Data loading**: we first load the JSON files containing the GutBrain datasets (platinum, gold, and development collections) and extract article texts, entity annotations, and relation annotations. We then initialize the `microsoft/BiomedNLP-BiomedBERT-base-uncased-abstract-fulltext` tokenizer to prepare the input for the neural network, inserting special entity markers—`[E1]. . . [/E1]` for subjects and `[E2]. . . [/E2]` for objects—into the raw text.
- **Preprocessing**: split the entities between the title and the abstract, and filter for relations involving entities from both.
- **Model configuration**: extend the BERT embedding layer to recognize the newly introduced entity markers. Instantiate the BiomedBERT model for sequence classification, where each input sequence (text with entity markers) is processed to determine the presence of a binary relation.
- **Training setup**: configure the AdamW optimizer and learning rate scheduler, and define the hyperparameters used to train the model, including number of epochs, batch size, learning rate, and sequence length.
- **Model training**: in each training step, perform a forward pass, compute the loss, backpropagate gradients, and update model weights via the optimizer and scheduler.
- **Validation**: after each epoch, evaluate model performance on the test split, reporting Micro and Macro Precision, Recall, and F1-score.

To predict relations between entities in the test dataset, we implemented the following pipeline. Starting from the named entities identified in Subtask 6.1, we generate all ordered pairs of distinct entities (e_1, e_2). We insert special markers (`[E1]`, `[E2]`) into the raw text to highlight the subject and object, respectively. The marked text is then tokenized into a fixed-length sequence and converted into BERT input tensors. We feed the encoded sequence into a fine-tuned `BertSequenceClassification` model, whose output logits corresponds to the relation labels defined in the `label2id.json` file. Finally, we apply the softmax function, select the top-scoring label with a confidence above 0.7, and save the predictions.

4. Experimental Setup

The experimental setup for this project includes the following components:

- The project source code is available in the `ataupd2425-gainer` repository on GitHub: https://github.com/Vezzero/ataupd2425-gainer.
- The dataset collection was provided by the CLEF BioASQ 2025 organizers and is available at: https://hereditary.dei.unipd.it/challenges/gutbrainie/2025/.
- The evaluation script used is `evaluation.py`, made available by the organizers via their official GitHub repository:
  https://github.com/MMartinelli-hub/GutBrainIE_2025_Baseline/blob/main/Eval/evaluate.py.
  Full details about provided data, baselines, and evaluation can be found in the overview paper [9].
- Model training and prediction for both NER and RE tasks were performed using the following hardware:
  - Tesla T4 GPU on Google Colab: https://colab.research.google.com/
  - Dual NVIDIA T4 GPUs on Kaggle: https://www.kaggle.com/
  - 8× NVIDIA A40 GPUs on DEI (Dipartimento di Ingegneria dell'Informazione) cluster: https://docs.dei.unipd.it/

A README file is provided in the GitHub repository with complete reproducibility instructions.

[12] https://pmc.ncbi.nlm.nih.gov/

Table 5
Description of the models used in the NER Subtask.

Model	Backbone	Training Dataset	Number of Steps	Batch Size	Maximum Token Length
PironA	NuNer_Zero	Platinum, Gold	6,000	2	356
PironD	NuNer_Zero	Platinum, Gold, Dev	8,000	2	600
PironS	NuNer_Zero	Platinum, Gold, Silver	12,000	2	600

Table 6
Results of the 6.1 Subtask.

Run ID	Training Dataset	Marco P.	Macro R.	Macro F1	Micro P.	Micro R.	Micro F1
ma	Platinum, Gold	**0.580758**	0.532186	**0.528085**	**0.833333**	**0.739693**	**0.783726**
md	Platinum, Gold, Dev	0.405397	0.541578	0.456943	0.639738	0.710590	0.673305
ms	Platinum, Gold, Silver	0.388916	**0.550468**	0.451079	0.633216	0.724333	0.675716

Table 7
Description of the models used in the RE Subtask.

Model	Backbone	Training Dataset	Number of Epochs	Batch Size	Learning Rate	Negative Ratio
PironBinaryTG	BiomedBERT	Platinum, Gold	5	8	2e-5	0.3
PironBinaryTGD	BiomedBERT	Platinum, Gold, Dev	8	10	1e-4	0.3
PironBinaryTGS	BiomedBERT	Platinum, Gold, Silver	8	12	1e-4	0.2
PironBinaryTGDNeuml	pubmedbert-base	Platinum, Gold, Dev	8	12	1e-4	0.2

5. Results

In this section, we report and describe the results obtained from the evaluation on the test data.

The runs submitted have been evaluated by the organizers on a held-out test set of 40 expert-annotated articles. These annotations are used as ground truth to compute the performance metrics.

5.1. Named Entity Recognition Results

For the NER Subtask, we trained three variants of the `numind/NuNER_Zero` model: PironA, PironD, and PironS. All models share the same architecture but differ in the training dataset, number of epochs, and training steps as reported in Table 5.

The NER test results are reported in Table 6. The PironA model with the **ma** run achieves the highest Micro-F1 score with high Micro-Precision and Micro-Recall. In particular, its Micro-Precision indicates that almost all entities are labeled correctly, resulting in the fewest overall errors. PironD with the **md** run obtains the highest Macro-Recall with lower Precision, suggesting it captures more true instances while generating more false positives. In contrast, PironS with the last run, **ms**, performs worst on both Micro and Macro scores, likely a consequence of the noisier annotations in the silver dataset.

5.2. Relation Extraction Results

For the RE Subtasks, we trained four different models, finetuning the `BiomedNLP-BiomedBERT-base-uncased-abstract-fulltex`[13] and `NeuML/pubmedbert-base-embeddings`[14]. BiomedBERT was fine-tuned under three different configurations, each varying only in training data and hyperparameters. Table 7 reports the models setup.

As explained in Section 3, we base our RE models on the entity spans produced by our three NER variants (**ma, md, and ms**). For each test document, we first apply one of these variants to identify and save entities and then feed them into the following RE model to generate the final relation labels.

[13] https://huggingface.co/microsoft/BiomedNLP-BiomedBERT-base-uncased-abstract-fulltext

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

Table 8
Performance metrics of the Binary Tag-Based RE Subtask.

Run ID	Training Dataset	Macro P.	Macro R.	Macro F1	Micro P.	Micro R.	Micro F1
ba1	Platinum, Gold	0.199798	0.398256	0.250001	0.41195	0.56710	0.477231
ba2	Platinum, Gold	0.193153	0.402862	0.245832	0.40361	0.58009	0.476021
ba	Platinum, Gold	0.238069	0.367984	0.269853	0.48092	0.54546	0.511156
bd1	Platinum, Gold, Dev	0.175111	0.540041	0.249663	0.32323	0.69264	0.440771
bd2	Platinum, Gold, Dev	0.169758	0.546031	0.244914	0.31262	0.69697	0.431635
bd	Platinum, Gold, Dev	0.215928	0.540124	0.290181	0.38517	0.69697	0.496148
bp1	Platinum, Gold, Dev	0.281089	0.365235	0.295995	0.50000	0.51948	0.509554
bp2	Platinum, Gold, Dev	0.251913	0.378696	0.285441	0.47727	0.54546	0.509091
bp	Platinum, Gold, Dev	**0.317064**	0.325444	**0.296847**	**0.61497**	0.49784	**0.550239**
bs1	Platinum, Gold, Silver	0.173007	0.542975	0.246921	0.31262	0.69697	0.431635
bs2	Platinum, Gold, Silver	0.166350	**0.548190**	0.240690	0.30112	**0.70130**	0.421326
bs	Platinum, Gold, Silver	0.213547	0.538597	0.287483	0.37915	0.69264	0.490046

5.2.1. Binary Tag-Based RE Results

Table 8 reports both macro- and micro-averaged metrics for all submitted runs. The **bp** run, trained on the combined platinum, gold, and dev relation sets, with the BinaryTGDNeuml classifier and the NER predictions of the **ma** run, achieves the highest Macro F1 and Micro F1. We attribute this strong performance to its underlying BiomedBERT encoder: initially pre-trained on millions of PubMed abstracts and then fine-tuned in a sentence-transformers framework, it produces 768 dimensional embeddings that encode biomedical semantics. These more accurate representations help the model distinguish valid relations and invalid ones better and assign higher confidence scores to its predictions. The **bd** runs group, based on the NER prediction in the **md** run, shows a high Macro and Micro Recall but low Macro and Micro Precision, suggesting it is a Recall-oriented model. The **ba** group shows the opposite: higher Macro and Micro Precision but lower Macro and Micro Recall, suggesting it is a Precision-oriented model. The **bs** runs group had the lower Micro and Macro Precision (even if the Macro and Micro Recall are similar to the **bd** group). This shows again that the silver dataset's noise negatively influences the Precision value.

In the scatter plot, Figure 3, each point represents a run, with Micro Precision on the x-axis and Micro Recall on the y-axis. The Recall-oriented **bd**/**bs** runs group cluster in the top-left with high Recall and low Precision. The Precision-oriented **ba** group is located in the mid-left. The balanced ones, **bp** group, appears toward the bottom-right, corresponding to their highest Micro F1.

5.2.2. Ternary Tag-Based RE Results

For the Subtask 6.2.2, we leveraged our PironBinary models, updating the inference pipeline to predict and return relations between entities. Table 9 reports the performance of each run.

The **td** run, obtained using the PironBinaryTGD model and the **ma** run, achieves the highest Micro Precision, resulting in the highest Micro F1. This suggests that the run correctly predicted nearly all relations between entities. In contrast, the **ts** run shows the best score in Macro Precision, as both rare and common relations contribute equally to the final average, leading to the best Macro F1. This comparison highlights that PironBinaryTGD provides the highest overall accuracy in the **td** configuration, while PironBinaryTGS, in the **ts** setup with fewer positive examples, excels in achieving balanced performance across all relation classes.

5.2.3. Ternary Mention Based RE Results

Finally, Table 10 reports the performance of all 6.2.3 runs. The **ts** configuration, using the PironBinaryTGS model with the entities of the **ma** run, achieves the highest Micro and Macro F1, making it the best run overall.

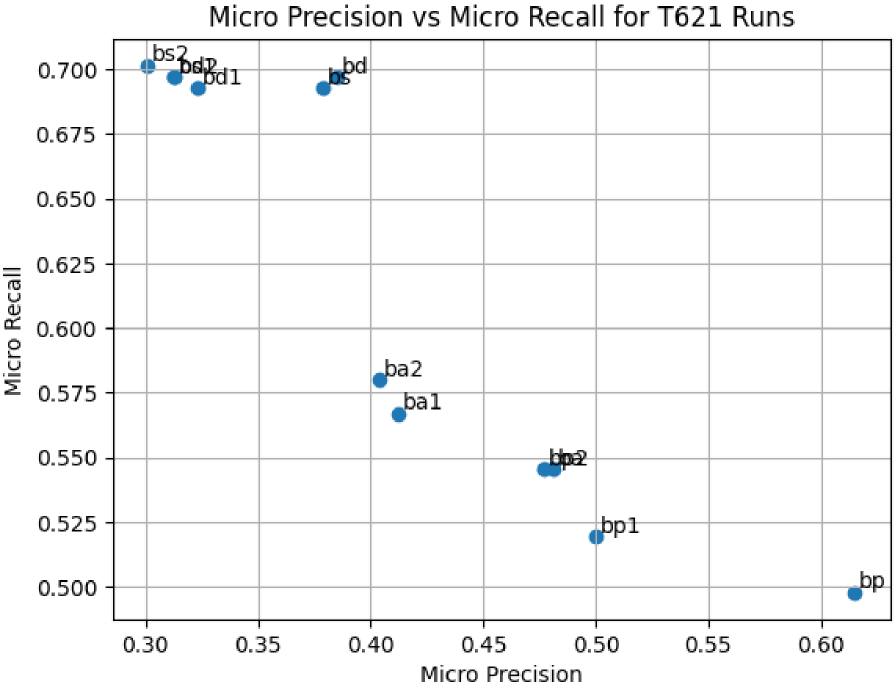

Figure 3: Scatter Plot of the Micro Precision and Recall of the 6.2.1 runs.

Table 9
Performance metrics of the Ternary Tag-Based RE Subtask.

Run ID	Training Dataset	Macro P.	Macro R.	Macro F1	Micro P.	Micro R.	Micro F1
ta1	Platinum, Gold	0.209531	0.178398	0.168514	0.581395	0.308642	0.403226
ta2	Platinum, Gold	0.209718	0.181461	0.168025	0.557971	0.316872	0.404199
ta	Platinum, Gold	0.254348	0.166327	0.181788	0.722772	0.300412	0.424419
td1	Platinum, Gold, Dev	0.286206	0.269013	0.259625	0.607362	0.407407	0.487685
td2	Platinum, Gold, Dev	0.271759	0.276622	0.252493	0.575419	**0.423868**	0.488152
td	Platinum, Gold, Dev	0.316739	0.231528	0.252816	**0.740458**	0.399177	**0.518717**
ts1	Platinum, Gold, Silver	0.271929	0.282908	0.258358	0.542373	0.395062	0.457143
ts2	Platinum, Gold, Silver	0.249736	**0.286284**	0.244771	0.500000	0.395062	0.441379
ts	Platinum, Gold, Silver	**0.322955**	0.257758	**0.272830**	0.681159	0.386831	0.493438

To explore how each run balances false positives and false negatives, Figure 4 plots Micro Precision against Micro Recall. In this scatter plot, the **tma** runs cluster in the upper left, with high Precision but low Recall, indicating they effectively avoid false positives but miss many true relations. The **tmd** group shows a well-balanced trade-off between Precision and Recall. The **tms** runs overlap with the **tmd** seeds, but the aggregated **tms** shifts into the top right quadrant, achieving the best combination of Micro Precision and Micro Recall—and consequently, the highest Micro F1.

6. Conclusions and Future Perspectives

In this paper, we presented the implementation pipelines for biomedical information extraction in the GutBrain corpus, covering both NER and RE. For Subtask 6.1, we fine-tuned the NuNer_Zero model under three hyperparameter configurations, achieving an almost 80% Micro F1 score. The chosen datasets used to train the model impacted the run performance, enabling us to see that the silver collections contained noisy annotations and leading to the worst results. In Subtasks 6.2.1, 6.2.2, and 6.2.3, we carried out RE as a marker-based sequence classification problem, using PironBinary models trained using the

Table 10
Performance metrics of the Ternary Mention-Based RE Subtask.

Run ID	Training Dataset	Macro P.	Macro R.	Macro F1	Micro P.	Micro R.	Micro F1
tma1	Platinum, Gold	0.106581	0.063951	0.070624	0.263666	0.109920	0.155156
tma2	Platinum, Gold	0.095498	0.062951	0.065099	0.238235	0.108579	0.149171
tma	Platinum, Gold	0.165853	0.071890	0.090573	0.422594	0.135389	0.205076
tmd1	Platinum, Gold, Dev	0.146181	0.123795	0.127507	0.294416	0.155496	0.203509
tmd2	Platinum, Gold, Dev	0.149414	0.122917	0.117869	0.262921	0.156836	0.196474
tmd	Platinum, Gold, Dev	0.212696	0.125437	0.141929	0.404984	0.174263	0.243674
tms1	Platinum, Gold, Silver	0.159961	0.134892	0.132749	0.285047	0.163539	0.207836
tms2	Platinum, Gold, Silver	0.133641	0.134407	0.119694	0.261242	0.163539	0.201154
tms	Platinum, Gold, Silver	**0.220283**	**0.138403**	**0.153766**	**0.427215**	**0.180965**	**0.254237**

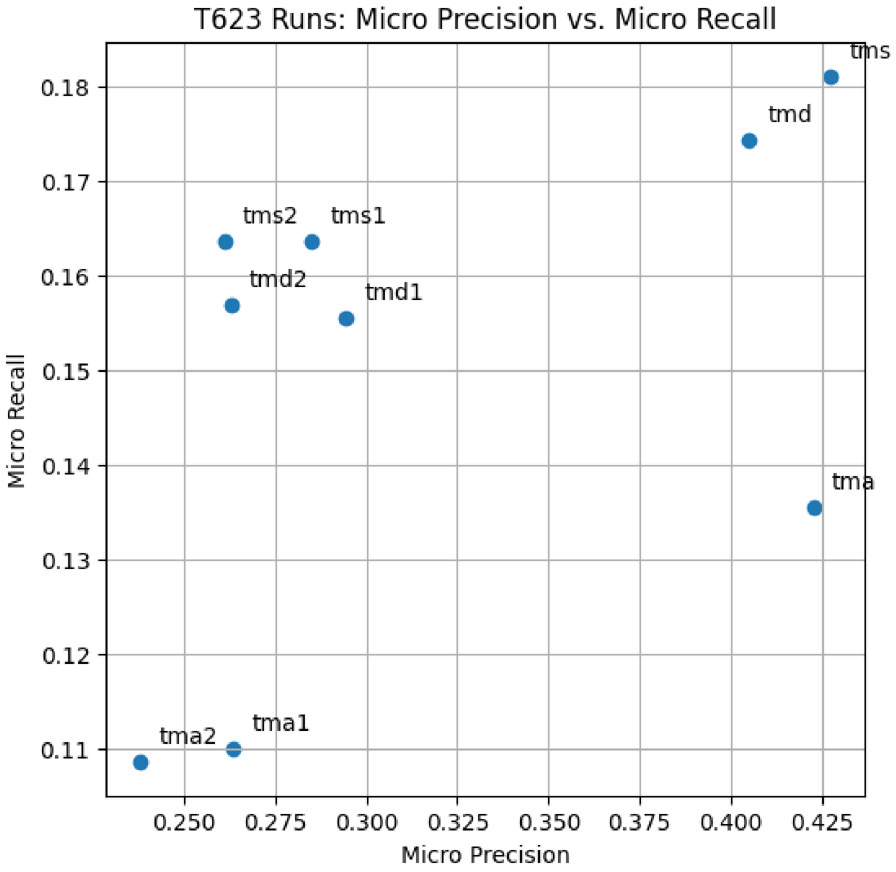

Figure 4: Scatter Plot of the Micro Precision and Recall of the 6.2.3 runs.

BiomedBERT base model. We were able to produce Precision-oriented, Recall-oriented, and balanced variants, each performing the best in different Subtasks, but without a dominating configuration.

Looking ahead, we plan to investigate and test new LLMs' models and more noise-robust training techniques, such as confidence-aware loss functions, to improve both NER and RE performance. In addition, we aim to integrate a semantic perspective grounded in linguistic analysis to enrich the linguistic and conceptual interpretation of extracted terms and relations. Specifically, we would like to apply semic analysis, which decomposes terms into minimal semantic units, as a structured approach to uncovering the internal organization of meaning in medical terminology [21, 22]. Incorporating this technique may enhance our ability to align terminological outputs with underlying conceptual structures, improving not only model interpretability but also the precision of information retrieval in domain-specific biomedical contexts.

Acknowledgments

This work is partially supported by the HEREDITARY Project, as a part of the European Union's Horizon Europe research and innovation programme under grant agreement No GA 101137074.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] A. Dunn, J. Dagdelen, N. Walker, S. Lee, A. S. Rosen, G. Ceder, K. Persson, A. Jain, Structured information extraction from complex scientific text with fine-tuned large language models, 2022. URL: https://arxiv.org/abs/2212.05238. arXiv:2212.05238.

[2] K. M. S. Islam, A. S. Nipu, J. Wu, P. Madiraju, Llm-based prompt ensemble for reliable medical entity recognition from ehrs, 2025. URL: https://arxiv.org/abs/2505.08704. arXiv:2505.08704.

[3] X. Zhao, Y. Deng, M. Yang, L. Wang, R. Zhang, H. Cheng, W. Lam, Y. Shen, R. Xu, A comprehensive survey on relation extraction: Recent advances and new frontiers, 2024. URL: https://arxiv.org/abs/2306.02051. arXiv:2306.02051.

[4] T. Nayak, N. Majumder, P. Goyal, S. Poria, Deep neural approaches to relation triplets extraction: a comprehensive survey, Cognitive Computation 13 (2021) 1215–1232. URL: http://dx.doi.org/10.1007/s12559-021-09917-7. doi:10.1007/s12559-021-09917-7.

[5] X. Chen, N. Zhang, L. Li, S. Deng, C. Tan, C. Xu, F. Huang, L. Si, H. Chen, Hybrid transformer with multi-level fusion for multimodal knowledge graph completion, in: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '22, ACM, 2022, pp. 904–915. URL: http://dx.doi.org/10.1145/3477495.3531992. doi:10.1145/3477495.3531992.

[6] K. Luo, F. Lin, X. Luo, K. Zhu, Knowledge base question answering via encoding of complex query graphs, in: E. Riloff, D. Chiang, J. Hockenmaier, J. Tsujii (Eds.), Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 2185–2194. URL: https://aclanthology.org/D18-1242/. doi:10.18653/v1/D18-1242.

[7] J. A. Diaz-Garcia, J. A. D. Lopez, A survey on cutting-edge relation extraction techniques based on language models, 2024. URL: https://arxiv.org/abs/2411.18157. arXiv:2411.18157.

[8] 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.

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

[10] M. Mohamed, P. Gader, Generalized hidden markov models. i. theoretical frameworks, Fuzzy Systems, IEEE Transactions on 8 (2000) 67 – 81. doi:10.1109/91.824772.

[11] C. Sutton, A. McCallum, An introduction to conditional random fields, 2010. URL: https://arxiv.org/abs/1011.4088. arXiv:1011.4088.

[12] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, 2013. URL: https://arxiv.org/abs/1310.4546. arXiv:1310.4546.

[13] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, C. Dyer, Neural architectures for named entity recognition, 2016. URL: https://arxiv.org/abs/1603.01360. arXiv:1603.01360.

[14] I. Hendrickx, S. N. Kim, Z. Kozareva, P. Nakov, D. Ó Séaghdha, S. Padó, M. Pennacchiotti, L. Romano, S. Szpakowicz, SemEval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals, in: K. Erk, C. Strapparava (Eds.), Proceedings of the 5th International Workshop on Semantic Evaluation, Association for Computational Linguistics, Uppsala, Sweden, 2010, pp. 33–38. URL: https://aclanthology.org/S10-1006/.

[15] D. Zeng, K. Liu, S. Lai, G. Zhou, J. Zhao, Relation classification via convolutional deep neural network, in: J. Tsujii, J. Hajic (Eds.), Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, Dublin City University and Association for Computational Linguistics, Dublin, Ireland, 2014, pp. 2335–2344. URL: https://aclanthology.org/C14-1220/.

[16] J. Li, T. Tang, W. X. Zhao, J.-Y. Nie, J.-R. Wen, Pretrained language models for text generation: A survey, 2022. URL: https://arxiv.org/abs/2201.05273. arXiv:2201.05273.

[17] U. Zaratiana, N. Tomeh, P. Holat, T. Charnois, Gliner: Generalist model for named entity recognition using bidirectional transformer, 2023. URL: https://arxiv.org/abs/2311.08526. arXiv:2311.08526.

[18] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, Journal of Machine Learning Research 21 (2020) 1–67. URL: http://jmlr.org/papers/v21/20-074.html.

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

[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, 2020. arXiv:arXiv:2007.15779.

[21] V. Bonato, G. M. Di Nunzio, F. Vezzani, A Novel Approach to Semic Analysis: Extraction of Atoms of Meaning to Study Polysemy and Polyreferentiality, Languages 9 (2024) 121. URL: https://www.mdpi.com/2226-471X/9/4/121. doi:10.3390/languages9040121, number: 4 Publisher: Multidisciplinary Digital Publishing Institute.

[22] V. Bonato, G. M. Di Nunzio, F. Vezzani, Preliminary Considerations on a Systematic Approach to Semic Analysis: The Case Study of Medical Terminology, Umanistica Digitale (2021) 211–234. URL: https://umanisticadigitale.unibo.it/article/view/12621. doi:10.6092/issn.2532-8816/12621, number: 10.