GutHub: Tackling the GutBrainIE Task Through Entity Recognition, Disambiguation, and Relation Extraction

Notebook for the BioASQ Task GutBrainIE on Gut-Brain Interplay Information Extraction at CLEF 2026

Andreas C. Rydder[†], Dziugas Austys[†], Simon B. Christensen[†], Juan Manuel Rodriguez and Daniele Dell'Aglio

Department of Computer Science, Aalborg University, Aalborg, Denmark

Abstract

In this paper, we introduce GutHub, a comprehensive extraction pipeline developed for the CLEF 2026 GutBrainIE task, addressing Named Entity Recognition (NER), Named Entity Disambiguation (NERD), and Relation Extraction at both the mention (M-RE) and concept (C-RE) levels. The primary obstacle in this domain is navigating extreme class imbalances across a multi-tiered training corpus that ranges from massive, noisy auto-generated text to sparse, expert-curated data. To overcome this, GutHub employs a stacked transformer ensemble combining domain-specific and zero-shot models to mitigate label bias. For disambiguation, SapBERT dense retrieval paired with a Mini-LM reranker trained on the MS-MARCO dataset outperformed several domain-specific alternatives. For mention-level relation extraction we combined class-specific confidence thresholding, deterministic heuristic pruning, and negative sampling. We then mapped the mention-level relations to unique URIs to output the concept-level relations. On the official test set, the micro-F1 scores achieved for each subtask were 0.7893 (NER), 0.3497 (NERD), 0.4029 (M-RE) and 0.1295 (C-RE).

Keywords

Stacked Ensemble, Transformer Models, Biomedical Information Extraction, Natural Language Processing, Named Entity Recognition, Named Entity Disambiguation, Relation Extraction

1. Introduction

For decades, the gut and the brain have been treated as separate entities. However, recent breakthroughs have uncovered a complex, bidirectional relationship between them, which is referred to as the gut-brain axis. Understanding this interplay is essential for improving well-being and treating conditions such as Alzheimer's, Parkinson's disease, mood disorders, and chronic pain. Yet, keeping pace with these discoveries is a massive challenge [1, 2]. The PubMed database now contains over 39 million citations, with approximately 1.7 million new entries added in the past year alone [3]. To assist researchers in managing this data influx, natural language processing (NLP) offers a powerful solution.

Within this context, the laboratory BioASQ [4] actively does research within this specific field and hosts tasks such as GutBrainIE [5] leveraging transformer-based models to automatically extract critical insights from biomedical literature. The GutBrainIE task consists of four subtasks:

- **Subtask 6.1.1 - Named Entity Recognition (NER)**: The first task focuses on Named Entity Recognition, which identifies and classifies text spans within PubMed abstracts into one of 13 predefined labels. These entity mentions can range from single words to multi-word text spans representing specific biomedical concepts.
- **Subtask 6.1.2 - Named Entity Recognition and Disambiguation (NERD)**: NERD detects entity mentions, classifies their label, and maps each to a unique concept identifier, giving experts the precise semantic meaning of every entity.

CLEF 2026 Working Notes, 21-24 September 2026, Jena, Germany

[†] These authors contributed equally.

andreascr@outlook.dk (A. C. Rydder); dziugasaust@gmail.com (D. Austys); sbc2401@gmail.com (S. B. Christensen); jmro@cs.aau.dk (J. M. Rodriguez); dade@cs.aau.dk (D. Dell'Aglio)

0000-0003-4904-2511 (D. Dell'Aglio)

© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

- **Subtask 6.2.1 - Mention-Level Relation Extraction (M-RE)**: M-RE addresses Relation Extraction (RE), aiming to identify semantic connections between entities based on a schema of 17 distinct predicate relations.
- **Subtask 6.2.2 - Concept-Level Relation Extraction (C-RE)**: C-RE involves identifying relationships between the biomedical concepts themselves.

The challenge provides a training corpus comprising PubMed titles, abstracts, and annotations for each task. The data is organized into three quality tiers: Bronze, Silver, and Gold. The Bronze dataset is the most extensive, containing over 2,972 documents. However, as it was automatically annotated using fine-tuned GLiNER (NER) and ATLOP (RE) models, it exhibits the highest noise level. The Silver dataset serves as an intermediate tier, consisting of 1,310 documents annotated by 95 trained students under expert supervision. This tier is divided into the 2025 dataset (499 documents) and the 2026 dataset (811 documents). The primary distinction between them is that concept-level relations in the 2025 dataset were automatically generated, whereas the 2026 dataset was manually annotated by students. Finally, the gold dataset represents the ground-truth, comprising 639 documents manually curated by 13 expert annotators. A held-out test set, derived from the gold collection, is also provided for model evaluation.

The varying quality tiers of the GutBrainIE corpus present a significant challenge for information extraction. Relying solely on the expert-curated Gold dataset restricts the volume of training data, rendering models prone to overfitting and bias. Conversely, carelessly fine-tuning on the generated annotated Bronze dataset without proper reflection of the included risks could introduce noise and degrade extraction accuracy.

Furthermore, class imbalance within the annotations must be addressed. For instance, in the Gold dataset, a label such as 'DDF' heavily outnumbers labels like 'Food' (6,909 vs. 271 instances). When framing NER as a token classification problem, this extreme class imbalance creates a strong bias for fine-tuned models toward majority labels, degrading their ability to generalize and accurately classify underrepresented entities. Beyond the dataset's complexities, the rapid surge of specialized biomedical transformers makes choosing the optimal foundation overwhelming, requiring a systematic comparison for the selection of our extraction pipeline.

To address the challenge, this paper presents a multi-stage extraction pipeline. For NER, we mitigate the inherent biases of individual models by fine-tuning an array of biomedical transformers and fusing their predictions through a confidence-scaled XGBoost stacking ensemble. To disambiguate these entities, we employ a two-stage retrieval architecture that pairs SapBERT for dense candidate retrieval with a Mini-LM cross-encoder for semantic reranking. Finally, our RE approach tries to combat the class imbalance by employing hard negative sampling and loss weighting during training. During inference, we use class-specific thresholding and a heuristic pruning pipeline that deterministically removes invalid relationships, such as self-referencing and nested entity overlaps.

2. Related Work

The rapid advancement of NLP has significantly expanded the domain of biomedical information extraction. In this section, we review the foundational literature that informs our proposed approach.

2.1. Biomedical Language Modeling

Since the introduction of transformers, the landscape of NLP has fundamentally changed [6, 7, 8]. Specifically, bidirectional encoders such as BERT have achieved state-of-the-art performance across various NLP domains since their release. To address the domain-specific challenge of biomedical NLP, BioBERT was introduced as a model specialized for the biomedical field and pre-trained on a large corpus of PubMed abstracts and PMC articles [9]. While this model established new benchmarks for various biomedical NLP tasks, it has since been improved upon by more advanced domain-specific models.

The recent surge in new AI models can make selecting the right tool for specific tasks overwhelming. To cut through the noise, we have put together a comparison of the most compelling models. Specifically, models that are pre-trained on biomedical corpora and deliver strong performance in NER.

BioLinkBERT [10] treats the pretraining corpus as a graph of documents connected by citations and hyperlinks, rather than as a collection of independent texts. During pretraining, linked documents are placed within the same context window, enabling the model to learn cross-document relationships in addition to intra-document context. Although it was trained on the same 21GB PubMed corpus as PubMedBERT, this link-aware pretraining objective allows it to capture richer inter-document dependencies from the same underlying data.

BioClinical ModernBERT [11] builds on the ModernBERT [12] base model, an optimized variant of the BERT architecture. It underwent continued pre-training on 53.5 billion biomedical tokens drawn from 20 datasets. Unlike the 512-token context window of most encoder models, ModernBERT supports an 8,192-token context, substantially extending the amount of text it can process at once. The model also uses an expanded vocabulary of 50,368 tokens, compared to BERT's 30,000.

GLiNER-BioMed [13] is a pre-trained model of the GLiNER architecture [14], specifically optimized through training on extensive biomedical corpora. Unlike traditional NER models that are constrained by a fixed set of entity types, GLiNER-BioMed is optimized for zero-shot recognition. Rather than framing NER as a sequence-labeling task constrained to a fixed set of entity types, its architecture formulates entity extraction as a bidirectional matching problem, enabling it to identify entity types not seen during training at inference time without further fine-tuning.

BioELECTRA [15] differs from models such as BioBERT in two respects. Whereas BioBERT is first pretrained on general-domain text before adaptation to biomedical corpora[9], BioELECTRA is pretrained exclusively on biomedical data, yielding a stronger native representation of domain-specific terminology. In addition, rather than masked language modeling, BioELECTRA employs Replaced Token Detection, which uses two networks: a generator that substitutes selected tokens with plausible alternatives, and a discriminator that predicts, for every token in the sequence, whether it is original or replaced. Because the discriminator evaluates all tokens, the model learns from the entire input sequence.

PubMedBERT [16] is pretrained from scratch on PubMed abstracts and full-text articles using a domain-specific vocabulary. This vocabulary allows complex biomedical terms to be represented without fragmentation into uninformative subword units, providing a more accurate baseline for domain-specific tasks.

OpenMed NER [17] focuses on training efficiency rather than on introducing new model architectures. It adapts existing base models, such as PubMedBERT, using a two-step pipeline. First, it applies Domain-Adaptive Pre-Training (DAPT) on a medical corpus to acquire domain-specific language. It then employs Low-Rank Adaptation (LoRA) to fine-tune the model for the entity extraction task. Through LoRA, the framework freezes the underlying model and updates less than 1.5% of the total parameters.

Table 1
Comparison of Biomedical Language Models for the GutBrainIE 2026 Challenge

Model	Unique Innovation	Context Window	Zero-Shot capabilities
BioLinkBERT	Document-relation prediction (graph-based)	512 tokens	No
BioClinical ModernBERT	Expanded vocabulary size	8,192 tokens	No
GLiNER-BioMed	Bidirectional matching	512 tokens	Yes
PubMedBERT	Custom medical dictionary	512 tokens	No
BioELECTRA	Replaced Token Detection	512 tokens	No
OpenMed NER	Applied DAPT and LoRA	512 tokens	No

2.2. Fine-Tuning and Output Optimizations

In parallel to architectural advancements, Low-Rank Adaptation (LoRA) has emerged as an efficient fine-tuning methodology for transformer models [17, 18, 19]. Instead of updating all model weights during fine-tuning, which risks "catastrophic forgetting" [20], LoRA freezes the base model and routes new knowledge through low-rank adapter modules. By updating only a small fraction of the parameters, LoRA substantially reduces the computational cost of fine-tuning relative to full-parameter updates, enabling more extensive testing and hyperparameter optimization.

2.3. Ensemble Learning and Aggregation Strategies

The primary objective of ensembling is to mitigate individual model biases and reduce the risk of overfitting, yielding a system substantially more robust than any single-model baseline [21]. In the context of biomedical Information Extraction (IE), the literature highlights several ensembling strategies:

- **Majority voting:** This technique aggregates predictions from multiple heterogeneous sources by assessing the label that receives the majority of votes across the ensemble [22, 23].
- **Seed Averaging:** This method consists of training multiple identical model architectures using different random seed initializations and averaging their predictions to reduce variance, while producing a more stable output [22, 24].
- **Rule-Based Ensembling:** Historically, NLP on rule-based frameworks relied on grammar production and finite-state automata. This methodology has been adapted as a class-specific strategy that leverages the strengths of individual models by prioritizing the prediction of the model with the highest historical accuracy for a given label [25].
- **Stacking Generalization (Meta-Classification):** This technique employs a secondary machine learning model (a meta-classifier) to aggregate the outputs from diverse base architectures. Rather than relying on fixed voting rules, the meta-classifier uses the continuous probability distributions of the base models as its input feature matrix [26].

3. Data Analysis

This section will explore the four data collections provided by the organizers to inform our preprocessing decisions. Specifically, we analyze the distribution of entities and relations relative to document length and identify potential outliers. Furthermore, we investigate noise in the Bronze dataset and introduce a strategy for outlier detection.

3.1. Feature & Label Distributions

Figure 1 shows the distribution of class labels in all collections, revealing a heavy class imbalance. The dataset is heavily dominated by the *DDF* (Disease, Disorder, or Finding) (36.6%) and *chemical* (14.7%) classes, which together comprise over 50% of all annotations. In contrast, *food*, *gene* and *statistical technique* make up only 1.7%, 1.5% and 1.5% of the annotations, respectively. This imbalance could become a challenge for NER. Without intervention, we risk our model being biased towards the dominating classes, meaning it would achieve high accuracy on DDF and chemical entities but suffer from low recall on the underrepresented classes.

Figure 2 (left) shows the distribution of entity counts relative to document length (word count) across all collections. The words are counted by concatenating the article's title and abstract and splitting by whitespace. We observe that both the median entity-per-word rate (≈ 0.13) and the interquartile range are relatively similar across all four collections. This suggests that the overall rate of entities per word is relatively consistent across the collections. There are, however, some outliers in the Bronze, Silver, and Silver 2025 collections, but none exceed an entity-per-word rate of 0.30. Notably, the Gold collection has zero outliers and is bounded at ≈ 0.25 entities per word.

Figure 1: Overall distribution of entity classes. Percentages represent the frequency of each label across all combined collections.

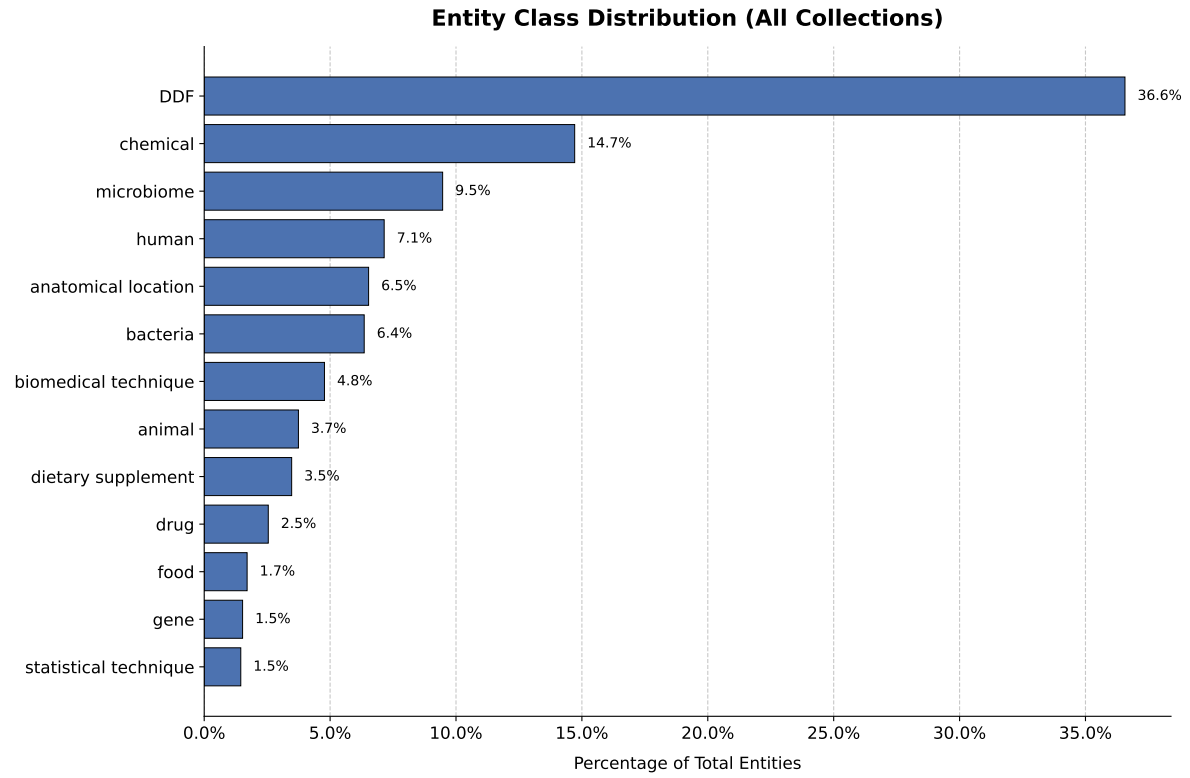
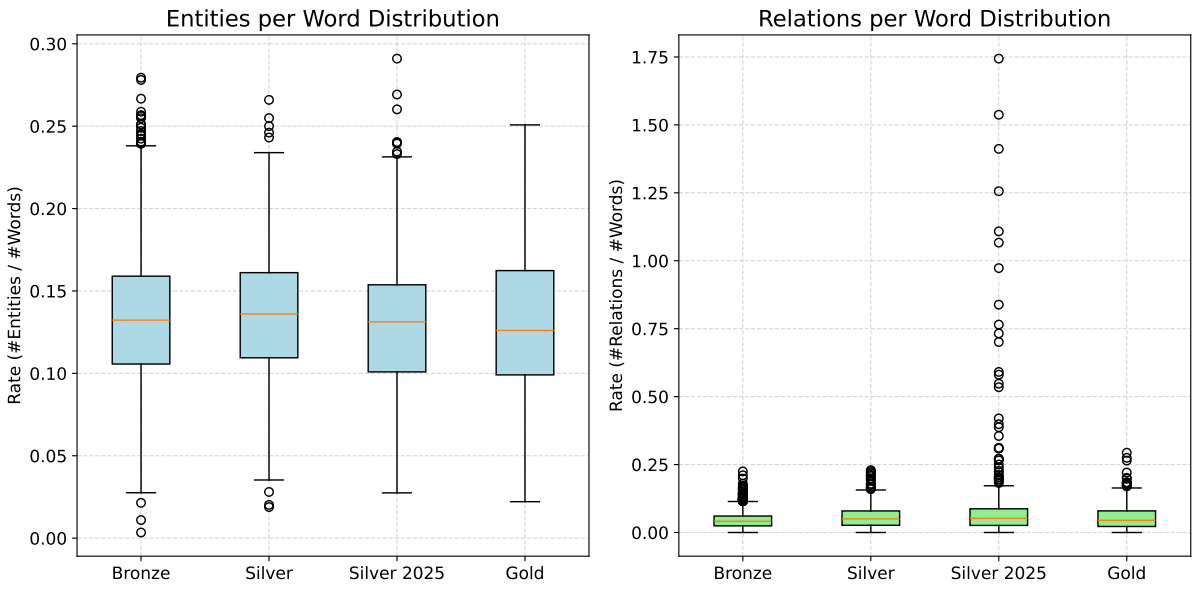


Figure 2: Distribution of entity counts (left) and relation counts (right) relative to article word counts, across the Bronze, Silver, Silver 2025, and Gold collections.



Conversely, looking at the relations-per-word rate, shown in Figure 2 (right), we observe higher variance. The Silver 2025 dataset features the most outliers, reaching up to 1.75 relations per word. In contrast, the Bronze, Silver, and Gold collections have tighter distributions with maximum outliers rarely exceeding 0.30. Despite the high variance in the Silver 2025 dataset, it is worth noting that the median relation-per-word rate is relatively low across the collections (≈ 0.05).

It can also be observed in Figure 2 (right) that a subset of articles across all collections contain zero

annotated relations. Specifically, in the Gold collection, 43 articles (corresponding to $\approx 6.7\%$) have no relations annotated.

3.2. Bronze Annotation Quality

We reviewed a sample of the AI-annotated Bronze dataset to examine whether it was subject to annotation noise and discovered issues with some labels. An example is the annotation for PubMed ID 35931825, which has "B" labeled as an anatomical location with a URI code that corresponds to dementia. These types of misclassifications demonstrate that the Bronze dataset is inherently noisy and should be used with caution. However, as the dataset makes up a significant part of the total data volume, completely discarding it would result in a significant loss of information. Consequently, we must introduce techniques that allow us to draw meaningful benefit from it despite its inherent noise.

3.3. Outlier Detection

The Gold dataset serves as our most reliable example of correct annotations and was therefore used to establish an upper boundary for outlier removal. To estimate the true population maximum for word, entity, and relation counts, we applied the minimum-variance unbiased estimator for a discrete uniform distribution. For each metric, the population maximum $\hat{N}$ was calculated using the formula:

$$\hat{N} = m + \frac{m}{k} - 1$$

where m is the maximum observed count in the Gold dataset and k is the total number of documents in the Gold sample ($k = 639$).

This calculation yielded maximum thresholds of 585 words, 84 entities, and 86 relations. We applied an OR condition to these criteria, meaning any document in the Bronze, Silver, or Silver 2025 collections that exceeded any of the three thresholds was flagged as an outlier.

Applying this threshold to the Bronze, Silver, and Silver 2025 collections isolated 25 outliers. A qualitative review of these instances revealed them to be primarily annotation anomalies, e.g., annotating the same relations for every synonym of a word on a document level. Consequently, these 25 documents were removed from the training corpus.

4. Methodology

In this section, we present the extraction pipeline to address the GutBrainIE task. The architecture operates sequentially, beginning with NER, where a meta-classifier extracts entity mentions. These mentions are then mapped to knowledge base concepts by the NERD module, using a two-stage dense retrieval and reranking approach. RE establishes document-level semantic connections between these entities, employing entity marking, class-specific thresholding, and heuristic pruning. Lastly, C-RE maps entities within these predicted relations to their corresponding URIs to output the final concept-level relationships.

4.1. Named Entity Recognition

Due to the specific problem domain and high complexity of the GutBrainIE corpus, relying on single model architectures limits extraction capabilities. The data distributions used to pre-train individual models can introduce inherent biases, concurrently creating blind spots on aspects with less training data. To overcome this performance hurdle, we chose to approach NER by fine-tuning a wide array of specialized base models and fusing their predictive distributions using a gradient boosting meta-classifier.

4.1.1. Base Models and Training Strategy

To capture a wider semantic representation of the text, we leverage a set of base architectures, including various biomedical BERT variants (PubMedBERT [16], BioLinkBERT [10], BioELECTRA [15], BioClinical ModernBERT [11], and OpenMed-NER [17]) alongside the zero-shot generalized model GLiNER [14, 13].

We framed NER as a token classification task. To reduce the usage of computational and temporal resources for full fine-tuning and to establish a robust baseline to track improvements, we initially evaluated each pre-trained model using the "Linear Probing" technique [27]. In this approach, the entire base network is frozen to preserve its original weights, and only a newly initialized classification head is trained to map the contextual embeddings to the target labels. The primary objective of this phase was to evaluate how well the out-of-the-box models could adapt to the highly specific entity distributions of the GutBrainIE corpus. This isolated evaluation provided a concrete baseline to determine the necessity, and eventual impact, of deeper full-network fine-tuning.

Furthermore, NER introduced alignment challenges during inference. Models utilizing Byte-Pair Encoding (BPE) [28] or WordPiece [29] tokenizers frequently absorb leading or trailing whitespace into subword tokens. Often, this can lead to character index misalignment when mapping from predicted tokens back to the original text. To avoid any alignment errors when using CLEF evaluation metrics, we implemented a dynamic boundary correction algorithm during post-processing, which strips whitespaces from the decoded text spans and calculates the absolute character start and end indices, ensuring precise alignment with the originally set document boundaries.

4.1.2. Ensemble

We initially considered rule-based majority voting ensembles, as it is a popular approach for multi-model architectures. However, static voting systems struggle to account for the varying domain expertise of different architectures. For example, one base model may be trained on specific data and excel at extracting complex chemical compounds, while another shows strong metrics in recognizing disease entities. Because majority voting treats all predictions with equal weight in a democratic manner, it works well for generalized models but effectively handicaps the impact of highly specialized architectures. [30]

To address this, we developed a stacking ensemble utilizing an XGBoost Meta-Classifier [26]. It takes a feature matrix of base model predictions as its input and outputs the final entity label. To generate this feature matrix, we implemented a confidence-scaled one-hot encoding strategy. For any given text span, we extracted the inference prediction and its associated confidence score from each fully fine-tuned model. We one-hot encoded these predictions, but instead of passing a binary value for the chosen label, we inserted the model's exact confidence score. This creates a feature matrix where the values directly represent how certain each model was about its prediction.

To avoid training the meta-classifier directly on in-sample training data, we generated the feature matrix using base-model predictions on the development dataset. Before constructing the candidate set, we sanitized outputs from each individual model. Varying architectures utilize different tokenization schemes (e.g., Byte-Pair Encoding [28] or WordPiece [29]), and some models include leading or trailing whitespaces within their predicted character offset. We stripped these whitespaces and recalculated the absolute start and end indices. This standardization ensures that tokenization differences do not prevent models from aligning on the exact entity boundary.

The common set of candidate text spans was then constructed by taking the union of all these sanitized span boundaries predicted by any of the base models. To build the feature matrix, we structured the data such that every unique candidate span generated a single row, and every base model was assigned its own dedicated feature column. If multiple models predicted the same text span and indices, they populated the same row, inserting their confidence scores into their respective columns. However, if boundaries or text spans do not strictly match, we do not attempt to merge them. Instead, we place them in separate rows in the matrix. In cases where a base model did not predict the given candidate span, its feature matrix column for that row was assigned a confidence score of 0.0. By evaluating

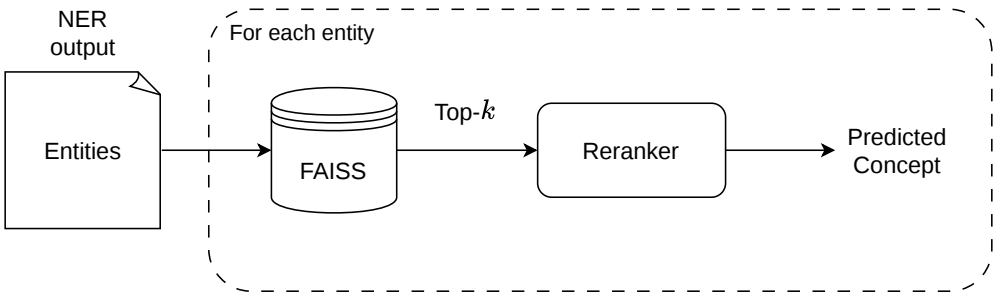
these confidence distributions against ground truth, the meta-classifier learned to perform final label assignment and span rejection by filtering out incorrect boundaries.

The XGBoost algorithm builds decision trees to dynamically learn the niche blind spots and patterns of each base model. To find the best configuration for the classifier, we used a random search technique to test different sets of hyperparameter combinations, such as maximum tree depth and learning rate. The motivation behind this was to find the settings that reach the highest overall micro-F1 score without memorizing the training data. The evaluation was done by using 5-Fold Cross-Validation [31] on out-of-sample development dataset, ensuring a robust baseline before generating final predictions on unseen test data.

4.2. Named-entity Recognition and Disambiguation

Given a document containing extracted text spans of named entities, the objective is to map each entity mention to a Uniform Resource Identifier (URI, also referred to as a concept) within a knowledge base. To achieve this, we adopt the retrieve-and-rerank paradigm that is well established for entity linking and dense retrieval [32, 33], in which a lightweight first-stage retriever narrows a large candidate space that is subsequently reordered by a more expensive second-stage model. The first stage consists of dense retrieval, where we use a bi-encoder to map entity mentions and knowledge base URIs into a shared vector space to efficiently fetch a broad set of likely candidates. The second stage performs reranking, applying a cross-encoder to evaluate and sort this smaller pool of candidates to determine the match. Figure 3 shows a visualization of the proposed architecture.

Figure 3: Overview of the proposed two-stage NERD architecture. For each entity extracted during the NER phase, the system first queries a FAISS index to perform dense retrieval, returning the top-k candidate concepts. In the second stage, a reranker evaluates this subset of candidates to determine and output the final predicted concept.



4.2.1. Preprocessing

Prior to inference, the knowledge base must be prepared. The organizers provided a comprehensive dataset consisting of text spans mapped to their corresponding URIs. For each URI labeled in the training data, we send a request to that URI and embed all the synonyms retrieved in the response. We initially hypothesize that including all training datasets in the knowledge base will yield the best performance by accommodating a larger vocabulary of entities. However, as established, the Bronze dataset is prone to noise, and we therefore evaluate whether including it helps or hinders retrieval accuracy in section 5.3. To construct the retrieval index, every text span in this dataset is passed through SapBERT [34] to generate dense vector representations. SapBERT is an SOTA (State of the Art) transformer model that is pretrained on the biomedical knowledge graph of UMLS [35]. These vectors then undergo L_2 normalization and are stored in a FAISS [36] index.

4.2.2. Dense Retrieval and Reranking

In the first stage, we use SapBERT to generate the embeddings for each entity mention. For a given entity mention text x, we process the sequence through the SapBERT tokenizer and model. We extract the representation of the `[CLS]` token from the final hidden layer and apply L_2 normalization to generate a dense vector embedding e. This query vector e is then evaluated against the pre-computed ANN index. The system retrieves the top-k candidate URIs and passes them to the reranking model.

For each retrieved candidate $c \in C_k$, we extract its textual name c_{name}. We concatenate the original entity mention x and the candidate text into a single sequence. The cross-encoder processes this sequence and returns a similarity score s_c:

$$s_c = \text{CrossEncoder}(x, c_{\text{name}})$$

The candidate that maximizes this cross-encoder score is selected as the final match. The corresponding URI for this optimal candidate c^* is then assigned to the entity mention:

$$c^* = \arg\max_{c \in C_k} s_c$$

4.3. Relation Extraction

Following the initial NER and NERD phase, the identified entities are passed to our RE pipeline. We formulate the RE task as a document-level sequence classification problem. Rather than classifying a single token, the model ingests the full document context containing the identified entity pairs and predicts a single, document-wide predicate to classify the relationship between them.

4.3.1. RE Preprocessing

Before performing RE, the data must be preprocessed to explicitly highlight the target entity pairs for the model. This is achieved using the entity marker technique [37]. Rather than the model evaluating the entire text uniformly, these markers force the Transformer's self-attention mechanism to focus on the specific relationship between the target pair. Specifically, for a given candidate pair, we surround the subject entity with `[E1]` and `[/E1]` tags, and the object entity with `[E2]` and `[/E2]` tags. Because this is a document-level relation extraction task, we generate every possible combinatorial pair of entities within the document, in both directions, and apply this entity marking technique to each pair.

4.3.2. Inference and Class-Specific Thresholding

Even with preprocessing and loss-weighting strategies during training, relying on a standard classification approach during inference, where the model outputs the class with the highest probability (the standard argmax), proved to be flawed for our case. Standard inference forces the model to commit to a prediction even under high uncertainty. In heavily imbalanced datasets, this behavior induces a large volume of false positives that significantly degrades overall performance.

To mitigate this, we implemented a class-specific decision threshold strategy. Rather than accepting the top prediction from the model, we extracted the per-class probabilities using a Softmax activation. A predicted predicate is accepted only if its probability exceeds a relation-specific threshold. Otherwise, the pair defaults to the "No Relation" label.

The thresholds were calibrated exclusively on the development set by inspecting the per-class precision and recall of each relation. The calibration followed a simple heuristic. Relations exhibiting high precision but low recall were assigned lower thresholds to admit more predictions, whereas relations exhibiting low precision but high recall were assigned higher thresholds to suppress false positives. Relations not requiring adjustment retained a default threshold of 0.80. The resulting thresholds are reported in Table 2.

Table 2
Class-specific confidence thresholds applied during RE inference, calibrated on the development set. Predicates not individually tuned use the default threshold of 0.80.

Predicate	Threshold
Compared to	0.99
Change abundance	0.99
Produced by	0.96
Is a	0.95
Located in	0.90
Interact	0.88
Target	0.85
Is linked to	0.85
Part of	0.85
Administered	0.85
Change effect	0.80
Affect	0.80
Impact	0.80
Change expression	0.80
Influence	0.75
Used by	0.75
Strike	0.65

This procedure deliberately trades recall for precision on the most error-prone relations. For example, "compared to" and "change abundance" initially exhibited near-zero and low precision, respectively. Assigning them a high threshold of 0.99 strongly suppresses their predictions, sacrificing recall on these classes in exchange for a substantial reduction in false positives that would otherwise dominate the micro-averaged score.

4.3.3. Post-Processing and Heuristic Pruning

While class-specific thresholding successfully mitigated some of the false positives, neural networks inherently lack logical reasoning capabilities and often struggle to resolve structural inconsistencies between entity pairs [38]. Due to this, the model occasionally predicted relations that passed the confidence threshold but were logically or structurally invalid within the context of the GutBrainIE corpus. To address this challenge, we implemented a deterministic post-processing pipeline that explicitly enforces semantic constraints over the neural network's outputs. We designed a pruning stage to remove these invalid predictions using the following heuristic rules:

- **Self-Referencing:** The model can predict interactions between an entity and itself, usually when the text spans were marked multiple times in a longer, complex sentence. Any predicted relation where the subject text string identically matched the object text string was discarded.
- **Nested Entity Overlap (NER Artifacts):** Due to artifacts originating from the NER stage, overlapping or nested entities would sometimes get passed to the RE pipeline. If the text span of one entity was completely contained in another entity's text span (e.g., "gut microbiome" and "microbiome"), the relation was removed. To prevent the deletion of short acronyms, this rule was enforced only on strings that are longer than three characters.

By applying these pruning rules, we deterministically removed impossible False Positives that evaded threshold checks and improved the micro-F1 score further.

4.3.4. Class Imbalance

The application of our entity marking technique described in section 4.3.1 resulted in a quadratic, $O(n^2)$, expansion of the dataset per document. This explosion introduced two significant challenges:

prohibitive computational costs during training and a severe class imbalance, as the vast majority of the generated candidate pairs did not possess a valid semantic relation. To address this, we implemented a two-stage reduction strategy.

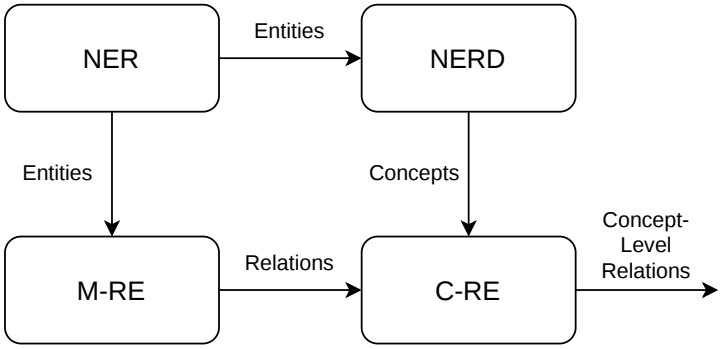
First, we utilized the strict dictionary of permissible entity interactions that was given by the organizers. We then said that any candidate pair whose entity types do not form a valid relation pathway (e.g., an interaction between a "bacteria" and a "biomedical technique") is immediately discarded. This rule serves as the primary filter, drastically reducing the initial search space by eliminating logically impossible samples.

Second, we used negative sampling on the negative candidate pairs. By artificially constraining the ratio of negative to positive candidate pairs, negative sampling effectively downsized the training data to a computationally feasible volume. To ensure the quality of negative samples, we applied a strict maximum distance of 60 words between entities. As 95% of valid relations fall within this threshold, this filter guarantees that our negative pool consists of "hard negatives" that share a realistic linguistic context. We note that this distance filter was intended to apply only to the negative pool. However, in our submitted system, it was inadvertently applied to the positive samples as well, excluding the small fraction of valid long-range positive pairs from training. We quantify the effect of this in Section 5.4.2. Furthermore, even after applying negative sampling with a 10:1 negative-to-positive ratio, the model was still too eager to default to the "No Relation" label, essentially ignoring the positive classes. To fix this, we had to introduce a strict penalty for missing true relations. By applying a 10x penalty multiplier to all positive labels and a 0.5x multiplier to the negative label, we forced the model to actively look for entity interactions and make positive predictions instead of over-predicting the negative class.

4.3.5. Concept-level Relation Extraction

The C-RE subtask serves as the final layer of our information extraction pipeline. M-RE provides the identified mention-level entity pairs alongside their predicted predicate label. The NERD pipeline provides the unique URIs mapped to those exact entity spans. C-RE relies on a deterministic mapping of mention-level relations to their corresponding URIs obtained from the NERD pipeline. By mapping the entities to their respective URIs, we generated concept-level tuples. These tuples were subsequently deduplicated per document to output only unique C-RE relations, since this was a requirement from the organizers.

Figure 4: System architecture for concept-level relation extraction.



5. Experiments

This section details the hardware setup and software environment, followed by a comprehensive performance evaluation of the NER, NERD, and RE pipelines on both the development and test sets. Unless stated otherwise, all results reported in this section are computed on the official development set,

for which ground-truth annotations were available to participants. The final test-set scores, computed solely by the organizers on the held-out data, are reported separately in Section 5.5.

5.1. Experimental Setup

All computational experiments were conducted on a high-performance mobile workstation. The hardware configuration included an Intel Ultra 9 275HX CPU, 32GB 6400MHz DDR5 RAM and an NVIDIA GeForce RTX 5080 16 GB VRAM GPU.

To ensure reproducibility, the software environment was standardized. The pipelines were implemented using PyTorch 2.11.0 for tensor operations. To natively support the Blackwell architecture of RTX 5080 GPU, hardware acceleration was managed via CUDA 12.8. Base language models were instantiated and trained using the HuggingFace transformers v4.51.3 library. The meta-classifier ensemble was constructed using xgboost 3.2.0. All other libraries and dependencies are disclosed in the project repository[1] under requirements.txt.

5.2. Named-entity Recognition

The evaluation of the NER pipeline is structured to reflect its iterative development, progressing from the establishment of pre-trained baselines to the full stacking ensemble.

5.2.1. Baseline Evaluation

To establish a baseline for our NER task, we selected language models pre-trained specifically on medical domain corpora. These models were evaluated on the dev dataset using the linear probing technique. Because standard pre-trained architectures are not tuned to output the specific dimensions required for a custom BIO (Begin, Inside, Outside) token classification scheme, direct zero-shot inference is structurally impossible. In this approach, we freeze the weights of each transformer to preserve their original pre-trained configuration. Only a newly initialized linear classification head is trained to map the frozen contextual embeddings to the target labels. The objective of this phase was to isolate and evaluate the capabilities of the base models without further interventions, establishing a concrete performance baseline for future fine-tuning and improvements. As detailed in Table 4, this approach yielded relatively low F1 scores across the classes. While the pre-trained models had an acceptable performance on standard entities such as “human” or “animal”, the predictions vastly degraded on more niche classes, such as “dietary supplement”, “food”, and “gene”. This baseline shows that while generic biomedical pre-training recognizes broad categories, the frozen embeddings are insufficient for this highly contextual task. Therefore, to improve the performance, the following phase transitions to full-network supervised learning.

Table 3
Performance of non-finetuned (left) and fine-tuned (right) biomedical models on the NER task on the development set measured in micro-F1. The highest score is highlighted in **bold**.

Model name	$F1_{micro}$
BioClinical ModernBERT Base	0.5076
BioClinical ModernBERT Large	0.5027
BioLinkBERT Base	0.6666
BioLinkBERT Large	0.6683
GLiNER Large	0.3362
OpenMed NER	0.6719
PubMedBERT	**0.6798**
BioELECTRA	0.6582

Model name	$F1_{micro}$
BioClinical ModernBERT Base	0.7467
BioClinical ModernBERT Large	0.7391
BioLinkBERT Base	0.7617
BioLinkBERT Large	0.7670
GliNER Large	0.5135
OpenMed NER	0.6903
PubMedBERT	**0.7698**
BioELECTRA	0.7440

[1] https://github.com/IllThinkOfSomething/Gut-Hub/tree/main

5.2.2. Fine-Tuned Models

Table 3 (right) presents the micro-F1 scores obtained after fine-tuning the models on the Bronze, Silver, Silver 2025, and Gold datasets. When comparing the results in Table 3 (left) and (right), it is evident that all models achieve substantially higher performance after fine-tuning. In particular, PubMedBERT improved from 0.6798 to 0.7698, while BioClinical ModernBERT Large increased from 0.5027 to 0.7467. These results demonstrate that domain adaptation through fine-tuning is essential for the GutBrainIE task. While the pretrained models possess general biomedical language understanding, they are not inherently optimized for the highly specialized entity distributions and annotation schema of the dataset. Fine-tuning enables the models to adapt to the task-specific terminology, contextual patterns, and label structures, resulting in significantly improved entity recognition performance.

5.2.3. Meta-Classifier Ensemble

While individual fine-tuned models achieved high scores, Table 4 shows that different architectures have different strengths. For instance, BioClinical ModernBert Large shows high prediction rates in "dietary supplements" and "microbiome" categories, while PubMedBERT is highly effective on the "anatomical location" label. To maximize the pipeline's capabilities and capitalize on each model's domain expertise, a meta-classifier ensemble was implemented using XGBoost. We built the feature matrix using a confidence-scaled encoding strategy: for each entity text span, the top prediction from every base model was one-hot encoded and multiplied by its corresponding confidence score. This created a matrix that allowed the meta-classifier to weight the base models based on their certainty for a specific entity class.

As shown in Table 5, combining all of the models into an XGBoost ensemble yielded our highest micro-F1 score of 0.8490. To validate whether the 8 model combination is required for the best results, a 3 model ensemble of the best performing architectures was evaluated (BioClinical ModernBERT Large, BioLinkBERT Large, PubMedBERT). This variation reduced the micro-F1 score to 0.8428. While it is still highly competitive, the performance gap showcases that even the lower-performing fine-tuned base models contribute to the overall ensemble score in a positive way. The XGBoost algorithm extracts useful predictions from weaker models, leveraging them for specific edge cases where dominant models fall short.

However, a methodological limitation must be noted regarding the evaluation of the ensemble. The feature matrix used to train the XGBoost meta-classifier was generated by running inference on the development dataset. Because the same development dataset was used during base model training to select the optimal checkpoints via the early stopping metric, the base models' predictions on this specific data represent an optimized, best-case scenario. While the base models were never directly trained on development data, using it for both checkpoint selection and meta-classifier training introduces an optimistic evaluation bias. Consequently, the reported ensemble micro-F1 scores likely reflect an upper bound of performance rather than a strict measure of out-of-sample generalization.

5.3. Named-entity Recognition and Disambiguation

To evaluate the best composition of data for the knowledge base, we experimented with different combinations of the Bronze, Silver, Silver 2025, and Gold datasets. The results of this experiment are shown in Table 6. We hypothesized that including the Bronze dataset would yield better results due to having an expanded vocabulary, and this is confirmed by the results. We also tried excluding both the Bronze and Silver 2025 datasets, but this led to a substantially lower micro-F1 score. Ultimately, combining the Bronze, Silver, Silver 2025, and Gold datasets yielded the highest overall performance. Consequently, we used this configuration for the final NERD knowledge base.

Following the knowledge base setup, we evaluated various candidate retrieval and reranking architectures for the NERD pipeline. The results of these configurations are detailed in Table 7. Using SapBERT to generate the embeddings for each predicted entity yielded a development-set micro-F1 of 0.4536.

Table 4
Label-specific NER performance across all categories on the development set, measured in F1. The highest score for each individual category is highlighted in **bold**. The table is truncated due to space constraints. The complete table is available at https://github.com/IllThinkOfSomething/Gut-Hub/blob/main/README.md. **Abbreviations**: BC MB (BioClinical ModernBERT), BLB (BioLinkBERT), and PMB (PubMedBERT).

Category	BC MB Large	BLB Large	PMB
DDF	0.8416	**0.8565**	0.8551
anatomical location	0.6943	0.7551	**0.7857**
animal	0.8447	0.8750	0.8679
bacteria	0.7056	0.7366	0.7259
biomedical technique	0.6519	0.6194	**0.6528**
chemical	0.5500	**0.6361**	0.6274
dietary supplement	**0.6528**	0.5278	0.5211
drug	**0.7485**	0.7059	0.7086
food	0.5333	0.4158	**0.6000**
gene	0.3167	0.3390	**0.4167**
human	0.8985	**0.9215**	0.9054
microbiome	**0.8694**	0.8655	0.8616
statistical technique	**0.7294**	0.6882	0.6585

Table 5
NER ensemble performance on the development set measured in micro-F1. The top row establishes the baseline. The middle row represents the performance of an ensemble combining all evaluated models. The bottom row displays the results of an ensemble consisting of the three models that achieved best individual performance. The highest score is marked in **bold**.

Ensemble	$F1_{micro}$
CLEF Baseline	0.7984
BioClinical ModernBERT Base + BioClinical ModernBERT Large + BioLinkBERT Base + BioLinkBERT Large + GliNER Large + OpenMed NER + PubMedBERT + BioELECTRA	**0.8490**
BioClinical ModernBERT Large + BioLinkBERT Large + PubMedBERT	0.8428

Table 6
Performance comparison of various dataset configurations for the NERD knowledge base on the development set measured in micro-F1. The highest score is marked in **bold**.

Datasets	$F1_{micro}$
Bronze + Silver + Silver 2025 + Gold	**0.4590**
Silver + Silver 2025 + Gold	0.4421
Silver + Gold	0.4124

To refine the retrieved candidates, we introduced a reranking stage using *cross-encoder/ms-marco-MiniLM-L6-v2* [2], a transformer model fine-tuned on the large-scale MS-MARCO passage ranking dataset. For brevity, we refer to this model as the MS-MARCO reranker. Adding this reranker further improved the micro-F1 to 0.4701.

Following this, we experimented with MedCPT [39], a cross-encoder trained on large volumes of PubMed biomedical literature and designed to handle domain-specific medical terminology. As shown

[2]https://huggingface.co/cross-encoder/ms-marco-MiniLM-L6-v2, Hugging Face, 2020.

in Table 7, MedCPT (0.4623) performed slightly worse than the MS-MARCO reranker. We hypothesize that this is because MedCPT was trained on long-form query–article pairs, specifically titles and abstracts. Because our NERD pipeline relies on matching short text spans between entity mentions and knowledge base concepts, this shift away from MedCPT's training distribution likely explains its lower performance.

Consequently, the combination of SapBERT embeddings and the MS-MARCO reranker was identified as the best-performing configuration for our system. This configuration reaches a development-set micro-F1 of 0.4701, which falls below the organizers' development-set baseline of 0.5949. We note, however, that the baseline drops substantially on the unseen test set, from 0.5949 to 0.4398, suggesting that development-set performance overstates generalization for the NERD subtask and motivating our focus on components that transfer to the held-out evaluation.

Table 7
Performance comparison of different setups for NERD on the development set measured in micro-F1. The highest score among our configurations is marked in **bold**.

Setup	$F1_{micro}$
CLEF Baseline	0.5949
SapBERT	0.4536
SapBERT + MedCPT reranker	0.4623
SapBERT + MS-MARCO reranker	**0.4701**

5.4. Relation Extraction

This section evaluates the design choices behind our relation extraction pipeline through experiments on the development set. We first analyze how threshold calibration and negative sampling jointly influence the balance between precision and recall, and then present a post-hoc ablation examining the effect of the 60-word entity-distance filter on the training data.

5.4.1. RE Threshold Calibration and Negative Sampling

The threshold values mentioned in Section 4.3.2 were determined by analyzing per-class precision on the development set. While stable relations retained the default threshold of 0.8, we enforced stricter confidence requirements for volatile labels to limit precision degradation. For example, predicates that initially suffered from severe false-positive rates, such as "interact" (initial precision of 0.0882) and "part of" (0.1429), improved to 0.1311 and 0.2000, respectively, after calibration.

Table 8 reports the contribution of each component to the overall micro-F1 on the development set. Starting from a 1:5 negative-sampling ratio (0.3479), adding heuristic pruning and class-specific thresholding raised the score to 0.3734. Our largest gain, however, came from increasing the negative-sampling ratio from 5 to 10, which alone raised micro-F1 to 0.4152. Interestingly, pruning contributed only marginally at this ratio (0.4152 to 0.4164). Combining the 1:10 ratio with pruning and class-specific thresholding produced our submitted configuration at 0.4667. This is the configuration evaluated on the official test set in Section 5.5.

5.4.2. RE Distance Ablation

As described in Section 4.3.4, we applied a 60-word maximum distance filter during the construction of the training data, motivated by the observation that roughly 95% of valid relations occur within this range. This filter was intended to constrain only the pool of hard negative samples, reducing training cost while preserving realistic negative contexts. However, in our submitted system the same filter was inadvertently applied to the positive samples as well, excluding the small fraction of valid long-range relations from the training data.

Table 8
Ablation of the RE pipeline components on the development set (micro-F1), showing the effect of the negative-sampling ratio, heuristic pruning, and class-specific thresholding relative to the CLEF baseline. The highest score is marked in **bold**.

Setup	$F1_{micro}$
CLEF Baseline	0.3910
Negative Ratio 1:5	0.3479
Negative Ratio 1:5 + Pruning + Threshold	0.3734
Negative Ratio 1:10	0.4152
Negative Ratio 1:10 + Pruning	0.4164
Negative Ratio 1:10 + Pruning + Threshold	**0.4667**

After submission, we conducted a post-hoc ablation to measure the effect of this constraint by retraining the pipeline with the distance filter removed entirely. As shown in Table 9, removing the filter increased the number of recovered true positives and substantially improved development-set micro-F1 from 0.4667 to 0.5561. This suggests that excluding long-range positive pairs during training measurably harmed the model's ability to recognize valid long-distance relations. As this experiment was performed after the official submission, the improvement is not reflected in our reported test-set results, and we highlight it as a clear direction for future work.

Table 9
Ablation comparing the RE pipeline with and without the 60-word distance limit, evaluated on the development set (micro-F1). The highest score is marked in **bold**.

Configuration	$F1_{micro}$
With 60-Word Limit	0.4667
No Distance Limit	**0.5561**

5.5. Official Test-Set Results

The experiments presented in the preceding subsections were conducted on the development set, for which ground-truth annotations were available. To assess the final performance of our pipeline under unbiased conditions, we submitted GutHub to the official evaluation on the held-out test set. The annotations for this set were never released to participants and were scored solely by the organizers, ensuring that the reported results are not subject to tuning or overfitting. Table 10 reports these results alongside the organizers' baseline.

On the test set, GutHub surpasses the baseline on M-RE, achieving a micro-F1 of 0.4029 compared to the baseline's 0.3893. This indicates that our combination of negative sampling, class-specific thresholding, and heuristic pruning provides a tangible benefit for identifying valid entity interactions on unseen documents. For NER and C-RE, our system performs comparably to the baseline, with micro-F1 scores of 0.7893 versus 0.7996 and 0.1295 versus 0.1348, respectively.

The largest gap appears in the NERD subtask, where GutHub reaches a micro-F1 of 0.3497 against the baseline's 0.4398. This is consistent with the development-set analysis in Section 5.3, in which our best retrieval-and-reranking configuration also fell below the corresponding baseline. Taken together, these results indicate that entity disambiguation is the subtask in which our pipeline has the most room for improvement, whereas M-RE is where our design choices yield the clearest advantage over the baseline.

We further observe that both GutHub and the baseline score consistently lower on the test set than on the development set. For instance, the baseline NERD micro-F1 decreases from 0.5949 on the development set to 0.4398 on the test set. This gap suggests that the test set poses a more challenging generalization problem for all systems.

Table 10
Official test-set results for GutHub compared against the organizers' baseline, measured in micro-F1. Test-set annotations were withheld from participants and scored solely by the organizers.

Sub-task	CLEF Test Baseline	GutHub
NER	**0.7996**	0.7893
NERD	**0.4398**	0.3497
M-RE	0.3893	**0.4029**
C-RE	**0.1348**	0.1295

6. Conclusion and Future Work

In this paper, we presented GutHub, an NLP pipeline specialized for the GutBrainIE challenge at CLEF 2026. Our system targets all four subtasks, specifically addressing the extreme class imbalance and varying data quality tiers present in the corpus. The key contributions and experimental insights from our pipeline include:

- **NER:** Using a population estimate derived from the Gold dataset, we identified and removed 25 outlier documents from the training corpus. We then deployed a stacking ensemble that combines the predictions of multiple base models, allowing the pipeline to exploit the label-specific strengths of each architecture. Through experimentation on ensemble composition, we found that including all available base models yielded the highest micro-F1 score.
- **NERD:** We experimented with different combinations of the training collections for constructing the knowledge base and found that including all datasets for URI extraction produced the best results. For the architecture, combining SapBERT dense retrieval with the MS-MARCO reranker gave the strongest configuration in our experiments, outperforming the domain-specific MedCPT reranker. We note, however, that our disambiguation results remained below the organizers' baseline, and we identify this subtask as the primary target for future improvement.
- **M-RE & C-RE:** To counter the quadratic expansion of candidate entity pairs, we integrated a dictionary of permissible interactions together with negative sampling bounded by a 60-word entity distance, reducing the candidate pairs to a feasible size for training. We further applied a 10x penalty multiplier to encourage positive predicate predictions. By combining an increased negative-sampling ratio with class-specific thresholding and deterministic post-processing pruning, we improved the development-set micro-F1 to 0.4627. The resulting mention-level relations were mapped to their corresponding URIs and deduplicated to produce the final concept-level relations.

On the official held-out test set, GutHub proved competitive with the organizers' baseline, exceeding it on M-RE (0.4029 vs. 0.3893) and performing comparably on NER and C-RE. Consistent with our development-set findings, NERD remained the subtask with the largest gap to the baseline. A detailed comparison is provided in Section 5.5.

Going forward, we aim to refine the GutHub architecture through iterative development and experimentation. For the NER pipeline, the primary objective is to address the evaluation bias identified in our current stacking methodology. Future iterations would construct out-of-fold predictions exclusively across the combined training tiers (Bronze, Silver, Gold). This adjustment would reserve the development dataset for final validation, providing a true measure of the pipeline's generalization capability. Beyond correcting the evaluation methodology, an open direction is to investigate additional ensemble architectures. Additionally, while our outlier detection strategy successfully isolated 25 qualitative anomalies, we have not yet empirically verified its impact on downstream performance. Future work will include ablation studies to quantify the benefit of this removal step. In RE, we plan to analyze the limitations of the current entity-distance boundaries, and to optimize the negative-sampling ratio and class-specific penalty weights for both positive and negative samples. For NERD, we will explore

training the cross-encoder directly, as our current results indicate promising potential. Collectively, these improvements are expected to enhance the system's overall accuracy and generalizability.

Declaration on Generative AI

During the drafting of this paper, Generative AI tools (specifically Gemini and DeepSeek) were utilized to refine language, academic tone, and structural flow. We have thoroughly reviewed, edited, and validated all AI-assisted text to ensure accuracy. All core research is original, and we assume full responsibility for the final document's content.

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.

References

[1] E. A. Mayer, K. Tillisch, A. Gupta, Gut/brain axis and the microbiota, Journal of Clinical Investigation 125 (2015) 926–938. doi:10.1172/JCI76304.

[2] S.-M. Petrut, A. M. Bragaru, A. E. Munteanu, A.-D. Moldovan, C.-A. Moldovan, E. Rusu, Gut over Mind: Exploring the Powerful Gut–Brain Axis, Nutrients 17 (2025) 842. doi:10.3390/nu17050842.

[3] E. W. Sayers, E. E. Bolton, A. M. Fine, C. Kelly, S. Kim, M. Landrum, S. Lathrop, A. Malheiro, T. D. Murphy, L. Phan, S. Pujar, B. W. Trawick, V. A. Schneider, K. D. Pruitt, Database resources of the National Center for Biotechnology Information in 2026, Nucleic Acids Research 54 (2026) D20–D27. doi:10.1093/nar/gkaf1060.

[4] A. Nentidis, G. Katsimpras, A. Krithara, M. Krallinger, M. Rodríguez-Ortega, E. Rodriguez-López, N. Loukachevitch, I. Rozhkov, 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 2026: The fourteenth BioASQ Challenge on Large-Scale Biomedical Semantic Indexing and Question Answering, Lecture Notes in Computer Science, Springer, 2026.

[5] 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. S. Salido, A. Barrón-Cedeño, A. G. S. de Herrera, S. MacAvaney, J. M. Struß (Eds.), CLEF 2026 Working Notes, 2026.

[6] B. Song, F. Li, Y. Liu, X. Zeng, Deep learning methods for biomedical named entity recognition: a survey and qualitative comparison, Briefings Bioinform. 22 (2021).

[7] N. Goyal, N. Singh, Named entity recognition and relationship extraction for biomedical text: A comprehensive survey, recent advancements, and future research directions, Neurocomputing 618 (2025) 129171. doi:10.1016/j.neucom.2024.129171.

[8] J. Li, A. Sun, J. Han, C. Li, A Survey on Deep Learning for Named Entity Recognition, IEEE Trans. Knowl. Data Eng. 34 (2022) 50–70.

[9] 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, Bioinform. 36 (2020) 1234–1240.

[10] M. Yasunaga, J. Leskovec, P. Liang, LinkBERT: Pretraining Language Models with Document Links, in: ACL (1), Association for Computational Linguistics, 2022, pp. 8003–8016.

[11] T. Sounack, J. Davis, B. N. Durieux, A. Chaffin, T. J. Pollard, E. Lehman, A. E. W. Johnson, M. B. A. McDermott, T. Naumann, C. Lindvall, BioClinical ModernBERT: A State-of-the-Art Long-Context Encoder for Biomedical and Clinical NLP, CoRR abs/2506.10896 (2025).

[12] B. Warner, A. Chaffin, B. Clavié, O. Weller, O. Hallström, S. Taghadouini, A. Gallagher, R. Biswas, F. Ladhak, T. Aarsen, G. T. Adams, J. Howard, I. Poli, Smarter, Better, Faster, Longer: A Modern Bidirectional Encoder for Fast, Memory Efficient, and Long Context Finetuning and Inference, in: ACL (1), Association for Computational Linguistics, 2025, pp. 2526–2547.

[13] A. Yazdani, I. Stepanov, D. Teodoro, GLiNER-BioMed: a suite of efficient models for open biomedical named entity recognition, Bioinform. 42 (2026). URL: https://doi.org/10.1093/bioinformatics/btag322. doi:10.1093/BIOINFORMATICS/BTAG322.

[14] U. Zaratiana, N. Tomeh, P. Holat, T. Charnois, GLiNER: Generalist Model for Named Entity Recognition using Bidirectional Transformer, in: NAACL-HLT, Association for Computational Linguistics, 2024, pp. 5364–5376.

[15] K. R. Kanakarajan, B. Kundumani, M. Sankarasubbu, BioELECTRA: Pretrained Biomedical text Encoder using Discriminators, in: BioNLP@NAACL-HLT, Association for Computational Linguistics, 2021, pp. 143–154.

[16] 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. Heal. 3 (2022) 2:1–2:23.

[17] M. Panahi, OpenMed NER: Open-Source, Domain-Adapted State-of-the-Art Transformers for Biomedical NER Across 12 Public Datasets, CoRR abs/2508.01630 (2025).

[18] I.-F. Gheorghita, V.-I. Bocanet, L. B. Iantovics, Fine-Tuning BiomedBERT with LoRA and Pseudo-Labeling for Accurate Drug–Drug Interactions Classification, 2025. doi:10.3390/app15158653.

[19] 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, in: ICLR, OpenReview.net, 2022.

[20] Y. Luo, Z. Yang, F. Meng, Y. Li, J. Zhou, Y. Zhang, An Empirical Study of Catastrophic Forgetting in Large Language Models During Continual Fine-Tuning, IEEE Transactions on Audio, Speech and Language Processing 33 (2025) 3776–3786. doi:10.1109/TASLPRO.2025.3606231.

[21] H. Zhang, M. O. Shafiq, Survey of transformers and towards ensemble learning using transformers for natural language processing, Journal of Big Data 11 (2024) 25. doi:10.1186/s40537-023-00842-0.

[22] W. L. Seow, I. Chaturvedi, A. Hogarth, R. Mao, E. Cambria, A review of named entity recognition: from learning methods to modelling paradigms and tasks, Artif. Intell. Rev. 58 (2025) 315.

[23] 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 (Working Notes), volume 4038 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2025, pp. 99–119.

[24] 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 (Working Notes), volume 4038 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2025, pp. 302–316.

[25] S. Taylor, C. Dil, A. Shah, Jannat, C. Oldham, A. Upadhyay, J. Varughese, N. Yazbeck, B. T. McInnes, NLP@VCU at BioASQ2025: Information Extraction on the GutBrainIE dataset, in: CLEF (Working Notes), volume 4038 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2025, pp. 617–625.

[26] T. Chen, C. Guestrin, XGBoost: A Scalable Tree Boosting System, in: B. Krishnapuram, M. Shah, A. J. Smola, C. C. Aggarwal, D. Shen, R. Rastogi (Eds.), Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016, ACM, 2016, pp. 785–794. URL: https://doi.org/10.1145/2939672.2939785. doi:10.1145/2939672.2939785.

[27] A. Kumar, A. Raghunathan, R. M. Jones, T. Ma, P. Liang, Fine-Tuning can Distort Pretrained Features and Underperform Out-of-Distribution, in: ICLR, OpenReview.net, 2022.

[28] R. Sennrich, B. Haddow, A. Birch, Neural Machine Translation of Rare Words with Subword Units, in: ACL (1), The Association for Computer Linguistics, 2016.

[29] M. Johnson, M. Schuster, Q. V. Le, M. Krikun, Y. Wu, Z. Chen, N. Thorat, F. B. Viégas, M. Wattenberg, G. Corrado, M. Hughes, J. Dean, Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation, Trans. Assoc. Comput. Linguistics 5 (2017) 339–351.

[30] A. Ghasemieh, A. Lloyed, P. Bahrami, P. Vajar, R. Kashef, A novel machine learning model with Stacking Ensemble Learner for predicting emergency readmission of heart-disease patients, Decision Analytics Journal 7 (2023) 100242. doi:https://doi.org/10.1016/j.dajour.2023.100242.
[31] R. Kohavi, A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection, in: IJCAI, Morgan Kaufmann, 1995, pp. 1137–1145.
[32] L. Wu, F. Petroni, M. Josifoski, S. Riedel, L. Zettlemoyer, Scalable Zero-shot Entity Linking with Dense Entity Retrieval, in: EMNLP (1), Association for Computational Linguistics, 2020, pp. 6397–6407.
[33] R. Nogueira, K. Cho, Passage Re-ranking with BERT, CoRR abs/1901.04085 (2019).
[34] F. Liu, E. Shareghi, Z. Meng, M. Basaldella, N. Collier, Self-Alignment Pretraining for Biomedical Entity Representations, in: NAACL-HLT, Association for Computational Linguistics, 2021, pp. 4228–4238.
[35] O. Bodenreider, The Unified Medical Language System (UMLS): integrating biomedical terminology, Nucleic Acids Research 32 (2004) D267–D270. doi:10.1093/nar/gkh061.
[36] M. Douze, A. Guzhva, C. Deng, J. Johnson, G. Szilvasy, P. Mazaré, M. Lomeli, L. Hosseini, H. Jégou, The Faiss Library, IEEE Trans. Big Data 12 (2026) 346–361.
[37] L. B. Soares, N. FitzGerald, J. Ling, T. Kwiatkowski, Matching the Blanks: Distributional Similarity for Relation Learning, in: ACL (1), Association for Computational Linguistics, 2019, pp. 2895–2905.
[38] Y. Ye, Y. Feng, B. Luo, Y. Lai, D. Zhao, Integrating Relation Constraints with Neural Relation Extractors, in: AAAI, AAAI Press, 2020, pp. 9442–9449.
[39] 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, 2023.