Trusting Gut Instincts: Transformer-Based Extraction of Structured Data from Gut-Brain Axis Publications

Notebook for the GutBrainIE Task of the BioASQ Lab at CLEF 2025

Lasse Ryge Andersen[1,†], Mikkel Hagerup Dolmer[1,†], Marius Ihlen Gardshodn[1,†], Juan Manuel Rodriguez[1] and Daniele Dell'Aglio[1]

[1]*Department of Computer Science, Aalborg University, Aalborg, Denmark*

Abstract

This paper presents the proposed solution by our team, Gut-Instincts, for the GutBrainIE task, which introduces a named-entity recognition (NER) subtask and three relation extraction (RE) subtasks on biomedical articles related to the gut-brain axis. To address the domain-specific terminology involved in the tasks, we rely on biomedical pretrained transformer-based models. For NER, we extend these with three different classification heads: (1) a dense layer, (2) a dense layer followed by a conditional random field (CRF), or (3) a bidirectional long short-term memory layer followed by a CRF. For RE, we introduce negative samples and experiment with different ratios between positive and negative samples. For all subtasks, we use model ensembling to reduce variability and improve robustness. Furthermore, since the provided dataset is of different quality levels, we use weighted training that enables the models to utilize all available data, while ensuring that high-quality data has a stronger influence during optimization. Our experimental results suggest that a large ratio of negative to positive samples, model ensembling, and weighted training improve performance in the NER and RE subtasks. In the GutBrainIE task, we placed second in the NER subtask (6.1) with an F1_{micro} score of 0.8382, and first place in all three RE subtasks 6.2.1, 6.2.2, and 6.2.3 with F1_{micro} scores of 0.6864, 0.6866, and 0.4635, respectively.

Keywords

Ensemble of Transformer-Based Models, Weighted Training, Named Entity Recognition, Relation Extraction, Biomedical Information Extraction, Gut-Brain Axis, Natural Language Processing

1. Introduction

In recent years, there has been a surge of research on the gut-brain axis, which refers to the intricate relationship between the gut and the brain. Once thought to affect only digestion, the gut microbiome is now recognized as a key component in neurological and mental health conditions [1]. Studies have revealed that imbalances in gut bacteria are linked to a range of conditions, including Alzheimer's disease, Parkinson's disease, anxiety, and depression [2].

As research in this domain intensifies, the volume of biomedical literature on the gut-brain axis has grown rapidly, presenting new challenges for researchers to keep up with the latest developments [3]. A recent analysis [4] revealed that PubMed has surpassed 37 million citations, with 1.6 million added in the past year alone. This makes it increasingly difficult to manually curate relevant findings. In response, there is an increased focus on developing information extraction systems that can support experts by automatically extracting and linking knowledge from scientific literature [3, 5, 6, 7, 8].

To support research on the gut-brain axis, the GutBrainIE task [9] is introduced, which is part of the BioASQ workshop [10]. The goal of the workshop is to "improve state-of-the-art methods from information retrieval, machine learning, natural language processing, and text mining"[1] for biomedical documents. In particular, the GutBrainIE task focuses on analyzing articles related to the gut-brain axis. To achieve this, the task presents four subtasks:

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

[†]These authors contributed equally.

lasse2507a@gmail.com (L. R. Andersen); mikkel@dolmer.dk (M. H. Dolmer); marius@gardshodn.com (M. I. Gardshodn); jmro@cs.aau.dk (J. M. Rodriguez); dade@cs.aau.dk (D. Dell'Aglio)

0009-0006-4874-1109 (L. R. Andersen); 0000-0002-1130-8065 (J. M. Rodriguez); 0000-0003-4904-2511 (D. Dell'Aglio)

[1] https://bioasq.org/about/objectives

- **Subtask 6.1 - Named-entity recognition (NER):** Classify specific text spans, referred to as entities, according to 13 predefined labels.
- **Subtask 6.2 - Relation extraction (RE):** Classify relations between entities through three distinct subtasks:
  - **Subtask 6.2.1 - Binary tag-based RE (BT-RE):** Identify which entity labels are related.
  - **Subtask 6.2.2 - Ternary tag-based RE (TT-RE):** Identify which entity labels are related and classify each relation according to 17 predefined labels.
  - **Subtask 6.2.3 - Ternary mention-based RE (TM-RE):** Identify which of the entities are involved in relations and classify each relation according to the 17 predefined labels.

When developing information extraction systems, a major cost is annotating the training dataset. According to Lawson et al. [11], the average cost of annotating entities in email datasets ranges from \$0.08 to \$0.22 per document using Amazon Mechanical Turk. However, this approach is not appropriate when annotating biomedical documents, as annotators need specialized domain knowledge to accurately perform the task.

Due to the high costs of annotating data and the expertise required to do so in the biomedical domain, it is often necessary to balance annotation quality with scalability. As a result, datasets of varying annotation quality are commonly created to support development. In this task, four datasets of different quality levels are provided:

- **Platinum dataset**: 111 articles with expert-curated annotations reviewed by external biomedical specialists.
- **Gold dataset**: 208 articles with expert-curated annotations.
- **Silver dataset**: 499 articles with annotations generated by trained students under expert supervision.
- **Bronze dataset**: 749 articles with automatically generated annotations, using fine-tuned models.

Additionally, the task also includes held-out development and test datasets, each consisting of 40 articles of the gold and platinum qualities.

As a result, the GutBrainIE task presents a wide range of obstacles. Firstly, biomedical articles are rich in specialized vocabulary, laden with ambiguous abbreviations, and often exhibit irregular grammar [12, 5]. As a result, general-domain language models are ineffective in this domain. Secondly, the training dataset has different quality annotations. Disregarding low-quality annotations reduces the size of the training set, which might lead to overfitting and not covering all the relevant cases. However, using the total amount of data without considering the quality could introduce too much noise in the training process.

This paper presents the proposed solution by our team, Gut-Instincts, for addressing the tasks of the GutBrainIE task. Our approach is an information extraction pipeline that uses ensembles of specialized models, built by extending well-known language models that have been pretrained on biomedical corpora [5, 6, 12]. We train each model using the provided datasets with a weighting scheme that reflects their quality, allowing for greater emphasis on higher-quality samples while still leveraging information from lower-quality samples. Our solution is available on GitHub[2].

Our experiments show that the proposed solution performs well across all four tasks. This is further supported by the official results in the GutBrainIE task, where our best NER ensemble resulted in second place for the NER subtask (6.1) with an F1_{micro} score of 0.8382 and our best RE ensembles achieved first place in the RE subtasks 6.2.1, 6.2.2, and 6.2.3 with F1_{micro} scores of 0.6864, 0.6866, and 0.4635, respectively.

This paper is structured as follows: In Section 2, we introduce related work on NER and RE in the biomedical domain. In Section 3, we explore and analyze the training datasets to uncover insights that are key in the development of our approach. In Section 4, we describe our approach to ensembling specialized models as well as the approach to training with a weighting scheme. In Section 5, we present

[2] https://github.com/P10-Natural-Language-Processing/Gut-Instincts

the experimental evaluation and key results for the final selection of the configurations of the models. Finally, in Section 6, we conclude the paper, presenting takeaways and open challenges.

2. Related Work

Since the introduction of BERT [13], transformer-based models have been the state-of-the-art technique for natural language processing. However, applying these advancements directly to biomedical information extraction yields unsatisfactory results due to a difference in vocabulary from general domain corpora to biomedical corpora [14]. Therefore, methods for pretraining and fine-tuning transformer-based models to perform specific tasks in the biomedical domain have been investigated [14, 5, 12, 6, 15]. One of the earliest models developed for this purpose is `BioBERT` [14], which extends BERT by further pretraining it on biomedical corpora. `BioBERT` outperforms the original BERT on many biomedical tasks.

After the success of `BioBERT`, the pretraining of transformer-based models on biomedical corpora has been further explored [12, 6, 16, 17]. `BioLinkBERT` [5] advances the pretraining approach by including information about links between documents, enabling the model to capture relationships across documents. This technique leads to improved performance over `BioBERT`. `BiomedBERT`[3] [6] is a BERT-based model trained from scratch exclusively on biomedical corpora, in contrast to `BioBERT`, which has initially been pretrained on general-domain corpora before being further pretrained on biomedical corpora. Finally, `BiomedELECTRA` [12] is also a model exclusively pretrained on biomedical corpora, but based on the ELECTRA pretraining strategy [18]. The ELECTRA pretraining strategy involves using discriminators to detect replaced tokens, while BERT-based models are trained to generate masked tokens. In summary, these models have achieved widespread success, which is highlighted in a recent survey [19] that identified over 30 publicly available variants. Therefore, we design our system to exploit the wide range of pretrained models.

NER and RE are core tasks of biomedical information extraction [20, 21, 22, 23, 24], enabling the conversion of unstructured biomedical text into structured knowledge. Despite the advancements in the field, challenges persist due to the complexity of biomedical terminology and the scarcity of annotated data, as annotation is time-consuming, costly, and requires domain expertise. However, despite the challenges, NER and RE are so relevant in the biomedical domain that dedicated benchmarks for evaluating biomedical language models have been introduced, such as the Biomedical Language Understanding and Reasoning Benchmark (BLURB) [6].

In state-of-the-art language models, NER is treated as a token-level classification task, typically accomplished by adding a classification layer on top of the output of the language model to map each token to an entity type or non-entity class [13]. This is the approach of models such as `BiomedBERT` [14] and `BiomedELECTRA` [12]. Prior to these models, it was common for NER to solely use a bidirectional long short-term memory (BiLSTM) combined with a conditional random field (CRF) [25]. The idea of using a CRF has been revived by either extending a language model with a dense layer followed by a CRF [26, 27] or a BiLSTM followed by a CRF [28, 27]. We consider these approaches while building our solution, experimenting with them to understand how they perform in the context of GutBrainIE.

Regarding the RE task, a simple approach is replacing the entities in a relation with special tokens representing the entity label and inputting the concatenated embeddings of these tokens to a classifier layer [6]. However, there is a risk of losing the specific information about the entities as they are replaced by generic tokens. Therefore, a widely adopted strategy to mark the entities in a relation is to insert special tokens around them [29, 30, 31, 32, 8, 33, 6]. Additionally, Baldini Soares et al. [29] explore different techniques for detecting relations between entities, including using the classification token of the model ([CLS]), pooling the embeddings of the tokens of the entities, and concatenating the embeddings of special entity start tokens. The latter technique exhibits the best performance.

Beyond architectural improvements, there are also other techniques for improving model performance. Bölücü et al. [34] and Wang et al. [35] propose methods for distinguishing between clean and noisy

[3] Previously known as `PubMedBERT`: https://huggingface.co/microsoft/BiomedNLP-BiomedBERT-base-uncased-abstract

samples and subsequently weighting the loss contribution of each sample based on its quality. Clean samples are given a higher weight, while noisy samples contribute less to the overall loss, thereby improving model robustness. We adopt a similar strategy to effectively leverage the data available in the four datasets introduced in Section 1.

Another technique is to ensemble models to improve performance [36, 37]. Model ensembles are effective in handling noisy or inconsistent data, as averaging predictions helps suppress errors from individual models. Additionally, ensembles enhance model stability and generalization, making them more reliable for real-world applications.

3. Exploratory Data Analysis

This section presents an exploratory data analysis that expands upon the initial analysis conducted by the organizers of the GutBrainIE task [9]. Specifically, the analysis examines the article lengths, entity and relation outliers, and annotation quality issues. These insights inform the development of our information extraction pipeline.

3.1. Token Sequence Length Distributions

We analyze the article lengths across all dataset quality levels by tokenizing the combined title and abstract of each article using WordPiece [13]. Figure 1 shows the token sequence length distributions. Overall, the distributions across all qualities appear to be close to normally distributed. The overall lengths are similar across the different qualities, as expected. Two articles are excluded from the figure: PubMed ID 37368331 with a token count of 1176 from the platinum dataset and PubMed ID 39299582 with a token count of 2924 from the silver dataset.

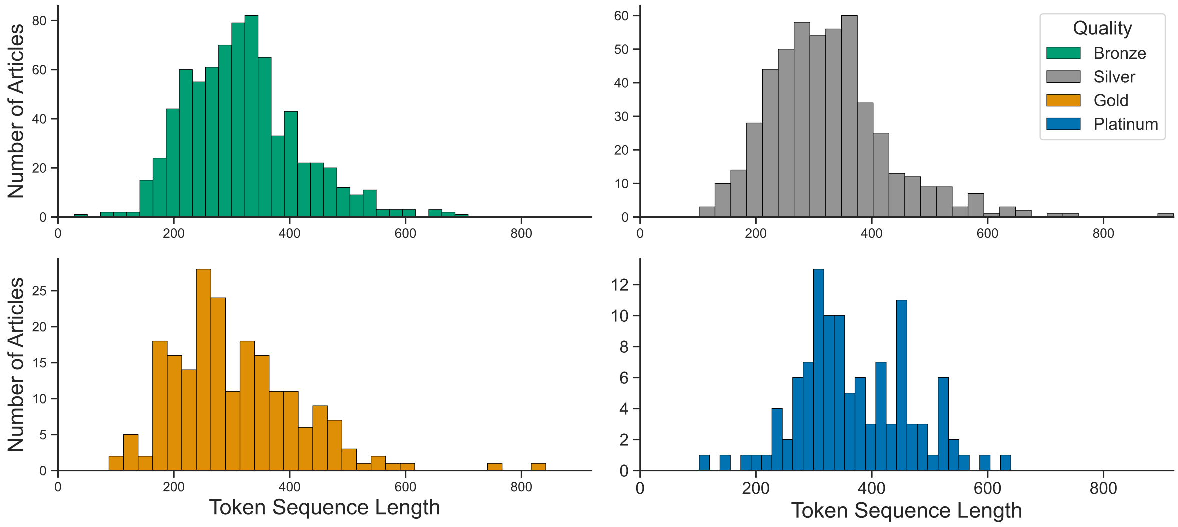
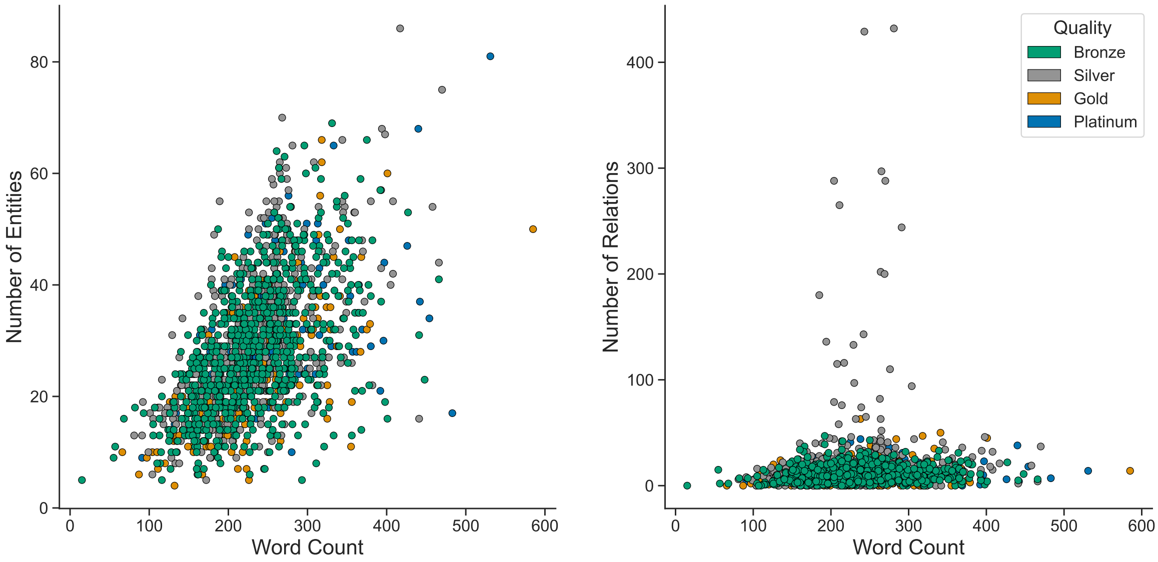


Figure 1: The token sequence length distributions across all dataset qualities of concatenated titles and abstracts, tokenized using WordPiece.

3.2. Article Outliers

Figure 2 shows the number of entities and relations as a function of word count for all articles across all dataset quality levels. As seen in Figure 2a, there is a general tendency towards more entities as the word count increases, which is expected. In Figure 2b, the number of relations remains below 60 for almost all articles, regardless of the word count. However, in the silver dataset, some articles exceed this threshold, with a few reaching over 400 relations. Such outliers can introduce noisy patterns that may negatively impact the ability of models to learn how entities relate to one another, potentially reducing overall performance.



(a) Number of entities vs. word count for all articles. (b) Number of relations vs. word count for all articles.

Figure 2: Entities and relations as a function of word count for all articles. The word count is determined by concatenating the title and abstract and splitting on whitespace.

3.3. Annotation Issues

The bronze dataset differs from the other datasets since the annotations are generated by a model rather than human annotators. As a result, it contains a higher level of noise. For example, the entity “,” is labeled as Biomedical Technique, and the entity “*, and progressive*” is labeled as DDF. These examples suggest that the model struggles with correctly annotating the articles. Furthermore, some articles from the bronze dataset contain very few entities and relations, with 21 articles having no relations at all. Including these in the training could introduce noise and negatively impact model performance. Despite these issues, we want to exploit the bronze dataset, as it makes up a significant amount of the total amount of data.

The silver dataset also includes some annotation issues besides the relation outliers previously mentioned. It includes instances where entity text spans are incorrectly annotated in singular form when the correct form should be plural. This does not adhere to the annotation guidelines and introduces inconsistencies.

We also observe that in all four datasets, there are HTML tags in many of the articles, including in the annotated text spans. We consider the HTML tags problematic because they can lead to an increased amount of noise in the training data since they are not used consistently and do not provide relevant context. During tokenization, the HTML tags will result in the generation of spurious tokens that do not represent meaningful content, leading to the model learning irrelevant patterns, which ultimately leads to a degradation in performance.

4. Methodology

This section presents our proposed solution to the GutBrainIE challenge, detailing the information extraction pipeline we developed. An overview of the pipeline is illustrated in Figure 3. It takes as input the titles and abstracts of biomedical articles about the gut-brain interplay. To address the NER subtask (6.1), the pipeline processes these articles using an NER ensemble that combines the predictions of multiple NER systems and outputs the final set of predicted entities. For the three RE subtasks (6.2.1, 6.2.2, and 6.2.3), the predicted entities along with the titles and abstracts of the original articles are passed to an RE ensemble, which similarly combines the predictions of multiple RE systems to output the final set of predicted relations. The information extraction pipeline outputs the predicted relations in a format corresponding to the specific RE task being addressed.

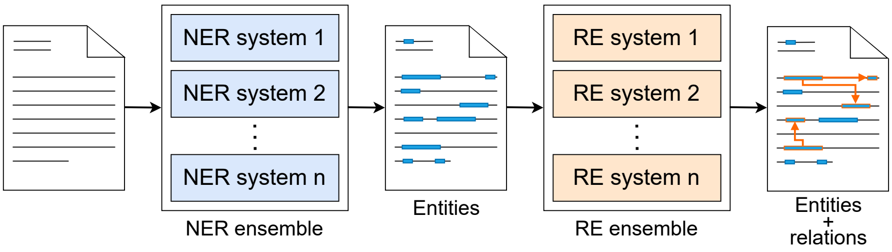


Figure 3: Information extraction pipeline overview.

4.1. Selection of Pretrained Models

For both the NER and RE tasks, we use pretrained models. The benefits of using pretrained models are the reduced computational cost from not training from scratch and the improved performance through transfer learning. We considered two selection criteria. Firstly, we considered models that are pretrained on biomedical corpora. This ensures that the models are better equipped to handle specific terminology and patterns encountered in the GutBrainIE task. Secondly, all of the selected models are uncased models, since the entities to be predicted in the NER task do not have a consistent casing, and therefore, including casing in the input representation would not provide a significant benefit and could introduce unnecessary variability.

Based on the criteria outlined above, we selected seven transformer-based language models:

- `BioLinkBERT-base` [5] and `BioLinkBERT-large` [5], which are pretrained on abstracts from PubMed[4]. Specifically, they are pretrained by feeding both a single document and its linked documents into the same language model context.
- `BiomedBERT-base-uncased-abstract-fulltext` [6], which is pretrained using abstracts from PubMed and full-text articles from PubMedCentral[5].
- `BiomedBERT-base-uncased-abstract` [6] and `BiomedBERT-large-uncased-abstract` [12], which are pretrained solely on PubMed abstracts.
- `BiomedELECTRA-base-uncased-abstract` [12] and `BiomedELECTRA-large-uncased-abstract` [12], which are also pretrained on abstracts from PubMed.

We also considered other pretrained models: `BioM-ALBERT-xxlarge` [15], `BioM-ALBERT-xxlarge-PMC` [15], and `BioBERT` [14]. However, the ALBERT models were excluded due to their substantially longer training times caused by their large size, and `BioBERT` was excluded because it consistently performed worse compared to the seven selected models.

4.2. Named-Entity Recognition (Subtask 6.1)

The first task in the information extraction pipeline is NER, which we approach as a token classification problem using the beginning-inside-outside (BIO) tagging scheme [7]. In this scheme, tokens that are part of an entity are labeled with one of two prefixes: B-, indicating the beginning of an entity, or I-, indicating a token inside an entity. Tokens that do not belong to any entity are labeled as O.

[4] https://pubmed.ncbi.nlm.nih.gov/
[5] https://pmc.ncbi.nlm.nih.gov/

In the NER ensemble shown in the information extraction pipeline overview in Figure 3, there are multiple NER systems. Each NER system consists of multiple components, which is illustrated in the NER system architecture in Figure 4. An NER system takes the titles and abstracts of biomedical articles as input, which are first preprocessed to ensure compatibility with the classification model. The classification model consists of a pretrained model extended with a classification head. The outputs of the classification model undergo a post-processing step, which converts them to entities, constituting the final output of the NER system. In the following, we detail the individual components of the NER system architecture, and then we discuss how we combine NER systems to create an NER ensemble.

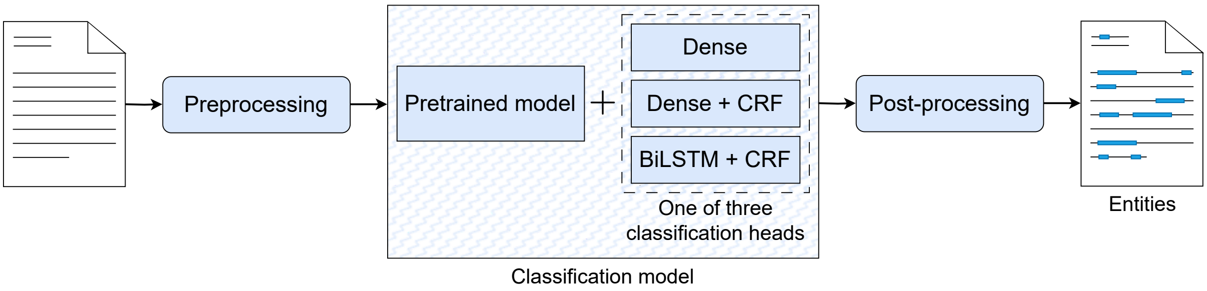


Figure 4: NER system architecture.

4.2.1. Preprocessing

The first component in the NER system architecture is the preprocessing of the input text. The input text is tokenized using WordPiece [13], which is a subword tokenization algorithm that splits words into smaller, frequent subword units based on a fixed vocabulary for the specific pretrained model. This allows the model to handle out-of-vocabulary words by representing them as combinations of known subwords. This capability is particularly important when performing NER on text from biomedical articles since the terminology contains a wide range of domain-specific expressions, including technical terms, abbreviations, and names of chemicals and genes.

The tokenization is performed with a maximum sequence length of 512 tokens. The resulting sequence of tokens is either padded or truncated to the maximum length to maintain compatibility with the architecture of the pretrained model. Given that only a small fraction of the dataset (77 out of 1567 samples) exceeds this limit, as shown in Section 3.1, this truncation minimally impacts the overall performance while ensuring compatibility with the architecture.

4.2.2. Classification Model

The second component in the NER system architecture is the classification model, which consists of a pretrained model (Section 4.1) extended with one of three classification heads:

- **Dense**: A single dense layer with size equal to the number of labels.
- **Dense + CRF**: A dense layer with size equal to the hidden size of the given pretrained model with a GELU activation function. This is connected to another dense layer whose size matches the number of labels, which feeds into a CRF.
- **BiLSTM + CRF**: A BiLSTM layer with hidden size set to half the hidden size of the given pretrained model with a GELU activation function. This is connected to a dense layer whose size matches the number of labels, which feeds into a CRF.

The output of the classification model consists of token-level predictions, each represented by a BIO label. The BIO labels are composed of the entity labels each prefixed with B- or I-. Tokens that do not belong to any entity are labeled as O.

4.2.3. Post-processing

The last component in the NER system architecture is the post-processing step, which transforms sequences of token-level predictions into entities. This step merges tokens into entities based on their predicted labels and their adjacency in the original input text. Two subsequent tokens t_1 and t_2, with start and end indices (s_1, e_1) and (s_2, e_2) in the original input text, are considered adjacent if $e_1 = s_2$ or $e_1 = s_2 - 1$. The token-level predictions are processed sequentially and merged into entities based on three rules:

1. A new span is initiated if the current token has a B- or I- prefix and there is no prior adjacent token with a B- or I- prefix.
2. Once a span is initiated, subsequent tokens can be appended to it if they are adjacent and share the same label, regardless of their prefix.
3. If there is no adjacent subsequent token or if it has a different label, then the current span is merged into an entity, and a new span is initiated.

Furthermore, we employ a lookahead mechanism to resolve rare cases, where the first and last tokens of a sequence of adjacent tokens share the same label, but the labels of the intervening tokens are different. Every time a span is initiated with a token with a B- prefix, the mechanism considers multiple subsequent tokens. If these are adjacent, have I- prefixes, and the last token has the same label as the initial token, then the labels of the intervening tokens are changed to the label of the initial token, allowing all considered tokens to be merged into a single entity.

When the token-level predictions are merged into entities, the casing of these entities is adjusted to match the casing of the original input text.

4.2.4. NER Ensemble

We consider two different NER ensemble methods to leverage the capabilities of multiple NER systems, to enhance the reliability of the model predictions. We name them entity-level ensemble and token-level ensemble.

Entity-level Ensemble The first method is an *entity-level ensemble*, which combines the outputs of individual NER systems at the entity-level, meaning this method is applied after the post-processing. The method involves an exact matching of the location of entities. Only entities for which the majority of participating NER systems agree are retained. Subsequently, the final label for each retained entity is determined based on a majority vote across the entity-level predictions from the participating NER systems.

Token-level Ensemble The second method is an *token-level ensemble*, which combines the outputs of individual classification models at the token-level prior to the post-processing. A prerequisite for this approach is to have consistent tokenization across all classification model outputs, ensuring that token-level predictions correspond to the same underlying input tokens across models.

This method involves a two-step process. In the first step, it is decided which token-level predictions are retained from the participating classification models. At each token position in the input sequence where at least one model predicts a label other than O, the predictions from all participating models are examined. The method adopts a token retainment strategy to determine whether the predictions at that position should be retained for the subsequent step. Three different token retainment strategies are proposed:

- **Union:** All token-level predictions from the participating models are retained regardless of agreement.
- **Majority:** Token-level predictions are retained only if the majority of the participating models have predicted a label other than O at the given position.

- **Intersection** Token-level predictions are retained only if all models assign a label other than O at the given position.

In the second step, the method determines the final label for each token position. Two different strategies are proposed for this purpose:

- **Softmax sum:** The softmax outputs of all the participating models are summed at the given token position. Then, the label corresponding to the highest score is selected as the final label.
- **Majority:** The label predicted by the majority of the models at a given token position is selected as the final label for the token at that position. In cases where labels receive an equal number of votes, the label is decided with the softmax sum strategy.

4.3. Relation Extraction (Subtask 6.2)

We approach the RE task as a sequence classification problem. The approach is inspired by the method proposed by Baldini Soares et al. [29], in which entity spans are explicitly marked in the input text using special tokens to enhance relation representation. Specifically, the special tokens `[E1]` and `[/E1]` are inserted around the subject entity, and `[E2]` and `[/E2]` around the object entity. The hidden states at the positions of the `[E1]` and `[E2]` tokens are extracted and concatenated to form a fixed-length vector, which is then passed through a dense classification head.

Each RE system in the RE ensemble shown in the information extraction pipeline overview in Figure 3 consists of multiple components. These are illustrated in the RE system architecture in Figure 5. An RE system takes as input the titles and abstracts of the original articles together with the predicted entities from the NER ensemble.

The structure of the RE system architecture is similar to that of the NER system architecture. The input is preprocessed to ensure compatibility with the model. The classification model consists of a pretrained model (Section 4.1) followed by a classification head. In contrast to the NER systems, which classifies each token in a sequence, the RE systems assign a single label to an entire sequence. Therefore, we exclusively use a dense classification head for the RE systems, as architectures with a CRF layer are more suitable for structured prediction tasks like NER. The classification model outputs relation predictions that are post-processed to produce the set of relations, which are the final output. In the following, we first describe the preprocessing and the post-processing components, and then we detail the RE ensemble method.

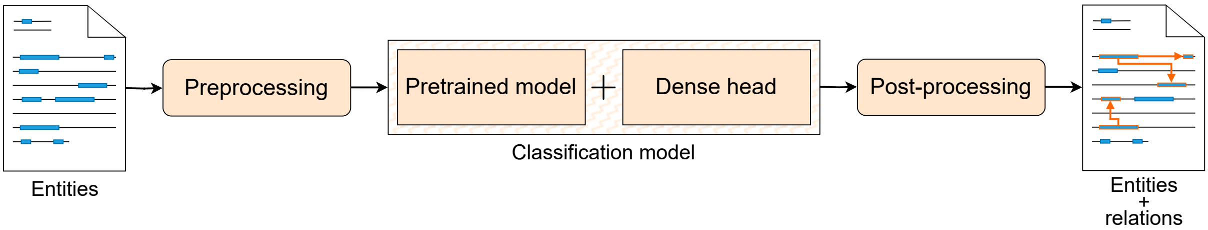


Figure 5: RE system architecture.

4.3.1. Preprocessing

The first component in the RE system architecture is the preprocessing of the input text. For each pair of entities, a separate sample is generated by inserting the special entity markers around the subject and object within the full input text. This ensures that each sample highlights exactly one pair of entities. Each sample is tokenized using the tokenizer of the given pretrained model, with the vocabulary extended with the four special tokens. The resulting sequence of tokens is either padded or truncated to the maximum length of 512 tokens to maintain compatibility with the architecture of the pretrained model.

4.3.2. Post-processing

The last component in the RE system architecture is the post-processing. After inference with the classification model, three post-processing steps are applied. The first step is to filter out any predictions where the classification model indicates that there is no relation between the subject and object, to retain only the meaningful relationships. The second step involves transforming each prediction into the format of the specific RE task. The third step identifies and removes duplicate predictions, ensuring that each relation is represented only once in the final output.

4.3.3. RE Ensemble

We propose a *relation-level ensemble* method to improve the robustness of the RE predictions. This method combines the outputs of multiple RE systems after post-processing and relies on exact matching of relations. For BT-RE (6.2.1), this requires the labels of both the subject and object to match. For TT-RE (6.2.2), the relation label is also included in the matching. For TM-RE (6.2.3), in addition to the relation label, the method further requires the exact text spans of the subject and object to match. Similarly to the entity-level ensemble strategy (Section 4.2.4), this method retains only those relations for which a majority of the participating models produce identical predictions.

4.4. Classification Model Training

To enhance performance, we construct a more realistic training dataset distribution using a negative sample multiplier for RE and incorporate weighted training to address differences in annotation quality across datasets.

4.4.1. Negative Sample Multiplier

For the RE tasks, to create a more realistic training dataset, the dataset is not limited to only containing the positive relations, which are the annotated relations from the datasets. We extend it with a configurable number of negative samples, which are instances of relations with random entities that are not related. We denote this configurable number as the negative sample multiplier. This parameter determines how many negative samples are added relative to the number of positive samples. We hypothesize that the negative sample multiplier influences model performance, and we study this effect in the experiments.

4.4.2. Weighted Training

We train the classification models for NER (Section 4.2) and RE (Section 4.3) using different combinations of the datasets of different qualities. We introduce a dataset weight vector, which is used as a hyperparameter, to account for the varying annotation quality across datasets:

$$\vec{w} = \begin{bmatrix} w_p \\ w_g \\ w_s \\ w_b \end{bmatrix},$$

where w_p, w_g, w_s, w_b denote the weights for the platinum, gold, silver, and bronze datasets, respectively. We associate each training sample to a weight based on its dataset quality. The weights are used during training to scale the loss contributions from individual samples according to the dataset quality of their source.

Let ℓ_i denote the loss for training sample i, and let $w(i)$ be the mapping from training sample i to its corresponding dataset weight. The weighted average training loss $\mathscr{L}_w$ over a batch of N samples is then computed as:

$$\mathscr{L}_w = \frac{1}{N} \sum_{i=1}^{N} w(i) \cdot \ell_i.$$

This approach enables the model to utilize all available data, including noisier datasets, while ensuring that higher-quality annotations have a stronger influence during optimization.

All models use the cross-entropy loss function, which is defined for training sample i as:

$$\ell_i = -\sum_{c=1}^{C} y_{i,c} \log(\hat{y}_{i,c}),$$

where C is the number of classes, $y_{i,c}$ is the true probability of class c, represented as 1 for the correct class and 0 for all other classes, and $\hat{y}_{i,c}$ is the predicted probability of class c.

5. Experiments and Results

In this section, the setup, experiments, and results are presented. Section 5.1, Section 5.2, and Section 5.3 detail the setup, dataset preparation, and training configurations, respectively. In Section 5.4, we analyze the experiments on the development dataset for the NER and RE subtasks. This includes exploring dataset weights, model architectures, and ensemble methods. In Section 5.5, the classification models of the best-performing systems are retrained. Then, the systems with the retrained models are evaluated on the test dataset and the results are shown. Finally, the final GutBrainIE leaderboard results are presented.

5.1. Setup

All training jobs are conducted on a high-performance computing cluster at Aalborg University, AI-LAB[6]. It uses the SLURM workload manager[7] to schedule and manage jobs, and leverages Singularity[8] to allow for creating and running containers. Each training job is allocated access to an NVIDIA L4 GPU with 24 GB VRAM, a 32-core AMD EPYC 7543 CPU, and 24 GB of RAM.

5.2. Dataset Preparation

We use all dataset qualities for training in the NER and RE subtasks. As the silver and bronze datasets contain some incorrect annotations, as described in Section 3.3, these are either corrected or removed.

Additionally, for all subtasks, we concatenate the title and abstract of each paper before tokenization to maintain a consistent input length and to accommodate RE, as relations can span from an entity in the title to an entity in the abstract.

In the bronze dataset, there are 21 articles with no relations at all, as detailed in Section 3.3. We exclude these articles from the training dataset for the RE subtasks.

As detailed in Section 3.2, the vast majority of articles contain fewer than 100 annotated relations, with only a small number of outliers in the silver dataset exhibiting more than 100 relations. We hypothesize that excluding these outliers enhances model performance. Therefore, we conduct experiments with and without the inclusion of these outliers.

5.3. Training Configurations

The training is performed using the AdamW optimizer [38] and a custom learning rate scheduler.

For the NER subtask, the models are trained for 20 epochs with a batch size of 16. The learning rate scheduler applies a linear warmup from $2 \cdot 10^{-5}$ at epoch 1 to $8 \cdot 10^{-5}$ at epoch 4, as using a warmup is proven to improve training stability and model performance [39]. From epoch 4 until epoch 13, the learning rate is kept constant at $8 \cdot 10^{-5}$. Then it decays with a factor of 0.8 every second epoch until finished.

[6] https://hpc.aau.dk/ai-lab/

[7] https://slurm.schedmd.com/

[8] https://docs.sylabs.io/guides/3.5/user-guide/introduction.html

For the RE subtasks, the models are trained for 10 epochs with a batch size of 16 using a similar scheduling strategy. A linear warmup is applied from 10^{-6} at epoch 1 to $4 \cdot 10^{-6}$ at epoch 3. From epoch 3 until epoch 6, the learning rate is kept constant at $4 \cdot 10^{-6}$. Then it decays with a factor of 0.5 every second epoch until finished.

The development dataset is used to validate the performance of each model after every epoch. The model state corresponding to the highest $\mathrm{F1}_{\mathrm{micro}}$ score is selected for each model. This metric is used because it is the official evaluation metric of the GutBrainIE task.

5.4. Experiments on the Development Dataset

This section presents the experimental results for the NER and RE subtasks on the development dataset.

5.4.1. Named-Entity Recognition (Subtask 6.1)

Impact of Dataset Quality Weighting To assess the impact of dataset quality on NER performance, the classification models are trained using various dataset weight combinations for the platinum, gold, silver, and bronze datasets. To ensure a fair comparison, the `BioLinkBERT-base` model with a dense classification head is used exclusively for this experiment. Table 1 presents the 18 dataset weight combinations considered, along with their corresponding $\mathrm{F1}_{\mathrm{micro}}$ scores on the development dataset. The best performance training with dataset weights achieves an $\mathrm{F1}_{\mathrm{micro}}$ score of 0.8314, whereas training without weighting achieves an $\mathrm{F1}_{\mathrm{micro}}$ score of 0.8189. The two dataset weight combinations yielding the highest $\mathrm{F1}_{\mathrm{micro}}$ scores are $\vec{w} = \begin{bmatrix}1.5 & 1.5 & 1 & 0.75\end{bmatrix}^\top$ and $\vec{w} = \begin{bmatrix}1.25 & 1.25 & 1 & 0.75\end{bmatrix}^\top$.

Table 1
The $\mathrm{F1}_{\mathrm{micro}}$, $\mathrm{precision}_{\mathrm{micro}}$, and $\mathrm{recall}_{\mathrm{micro}}$ scores on the development dataset for 18 dataset weight combinations using `BioLinkBERT-base` with a dense classification head. The two dataset weight combinations resulting in the highest $\mathrm{F1}_{\mathrm{micro}}$ scores are in bold.

$\mathbf{w_p}$	$\mathbf{w_g}$	$\mathbf{w_s}$	$\mathbf{w_b}$	$\mathbf{Precision_{micro}}$	$\mathbf{Recall_{micro}}$	$\mathbf{F1_{micro}}$
1.5	1.5	1.5	0.75	0.7866	0.8380	0.8114
1.5	1.5	1.5	0.5	0.7918	0.8478	0.8188
1.5	1.5	1.5	0.25	0.7920	0.8317	0.8114
1.5	**1.5**	**1**	**0.75**	**0.8157**	**0.8478**	**0.8314**
1.5	1.5	1	0.5	0.7888	0.8523	0.8193
1.5	1.5	1	0.25	0.7949	0.8397	0.8167
1.25	1.25	1.25	0.75	0.7911	0.8478	0.8185
1.25	1.25	1.25	0.5	0.8001	0.8424	0.8211
1.25	1.25	1.25	0.25	0.7923	0.8505	0.8204
1.25	**1.25**	**1**	**0.75**	**0.8003**	**0.8469**	**0.8230**
1.25	1.25	1	0.5	0.8015	0.8317	0.8163
1.25	1.25	1	0.25	0.7892	0.8344	0.8111
1	1	1	0.75	0.7913	0.8451	0.8173
1	1	1	0.5	0.7852	0.8380	0.8107
1	1	1	0.25	0.7938	0.8514	0.8216
1	1	1	1	0.7935	0.8460	0.8189
1	1	1	-	0.8064	0.8317	0.8189
1	1	-	-	0.7264	0.8228	0.8093

NER Classification Model Performance To investigate the performance across the three classification heads described in Section 4.2 and the seven pretrained models described in Section 4.1, we study three head variants of each of the seven pretrained models. This results in 21 classification models, which we train using the two best dataset weight combinations identified in Table 1. Table 2 shows the resulting $\mathrm{F1}_{\mathrm{micro}}$ scores.

The difference in $\mathrm{F1}_{\mathrm{micro}}$ scores across the three classification head categories is minimal. The classification models with dense heads perform slightly worse on average. However, although there are slight differences, the best-performing models are distributed across all three classification heads and different pretrained models. Hence, we do not observe a pattern where one classification head consistently outperforms the other two.

Moreover, the results do not reveal a pattern across dataset weight combinations or architectures, and therefore, no single architecture or classification head can be conclusively identified as the best.

Table 2
The $\mathrm{F1}_{\mathrm{micro}}$ scores on the development dataset of all combinations of pretrained models and classification heads under two dataset weight combinations. The two highest $\mathrm{F1}_{\mathrm{micro}}$ scores for each classification head are in bold.

Pretrained Model	**Dense**	**Dense + CRF**	**LSTM + CRF**
$\vec{w} = \begin{bmatrix}1.5 & 1.5 & 1 & 0.75\end{bmatrix}^\top$			
`BioLinkBERT-base`	0.8151	**0.8213**	0.8152
`BioLinkBERT-large`	0.8094	0.8185	0.8149
`BiomedBERT-base-uncased-abstract`	0.8103	0.8193	0.8092
`BiomedBERT-base-uncased-abstract-fulltext`	0.8118	0.8176	**0.8268**
`BiomedBERT-large-uncased-abstract`	0.8148	0.8170	0.8178
`BiomedElectra-base-uncased-abstract`	0.8050	0.8174	0.8123
`BiomedElectra-large-uncased-abstract`	0.8072	**0.8201**	0.8198
$\vec{w} = \begin{bmatrix}1.25 & 1.25 & 1 & 0.75\end{bmatrix}^\top$			
`BioLinkBERT-base`	**0.8217**	0.8166	0.8158
`BioLinkBERT-large`	0.8097	0.8085	**0.8202**
`BiomedBERT-base-uncased-abstract`	0.8019	0.8109	0.8077
`BiomedBERT-base-uncased-abstract-fulltext`	0.8067	0.8174	0.8152
`BiomedBERT-large-uncased-abstract`	0.7831	0.8183	0.8176
`BiomedElectra-base-uncased-abstract`	0.8013	0.8156	0.8100
`BiomedElectra-large-uncased-abstract`	**0.8211**	0.8165	0.8162
Average	**0.8085**	**0.8168**	**0.8156**

NER Ensemble Performance Finally, we investigate how the NER ensemble methods described in Section 4.2.4 perform. Token-level ensemble aggregates predictions for each token individually, whereas entity-level ensemble retains complete entities based on majority voting across models. Both strategies improve the performance, however, the entity-level ensemble approach consistently outperforms the token-level approach. Consequently, only entity-level ensembles are considered in the following experiment and for the submission.

To determine which number of NER systems yields the best performance for the entity-level ensemble approach, we conduct experiments with different sizes, ranging from 3 to 17 NER systems. For each n-ensemble size, we select the top-n NER systems based on the results of the previous experiments, ranked by their individual $\mathrm{F1}_{\mathrm{micro}}$ scores on the development dataset. As a result, each larger ensemble includes all NER systems from the smaller ensembles. For example, the top-5 ensemble contains all NER systems from the top-3 ensemble, including the next two highest-scoring NER systems. Table 3 shows the performance of each ensemble size on the development dataset.

The results indicate that ensemble strategies lead to a notable performance improvement compared to individual models. The highest $\mathrm{F1}_{\mathrm{micro}}$ score is observed with an ensemble size of 9. However, as performance remains consistently high across all ensemble sizes in Table 3, we consider all the ensemble sizes for the submission of the NER subtask (6.1).

Table 3
The F1_{micro}, $\text{precision}_{\text{micro}}$, and $\text{recall}_{\text{micro}}$ scores on the development dataset of ensembles of varying sizes. The result with the highest F1_{micro} is in bold.

Ensemble size	$\textbf{Precision}_{\textbf{micro}}$	$\textbf{Recall}_{\textbf{micro}}$	$\textbf{F1}_{\textbf{micro}}$
3	0.8351	0.8612	0.8480
5	0.8376	0.8585	0.8479
7	0.8360	0.8532	0.8445
9	**0.8436**	**0.8594**	**0.8514**
11	0.8398	0.8585	0.8490
13	0.8406	0.8594	0.8499
15	0.8405	0.8585	0.8494
17	0.8393	0.8559	0.8475

5.4.2. Relation Extraction (Subtask 6.2)

We conduct all RE experiments presented in this section using BT-RE classification models. Due to time constraints, the configurations yielding the best performance in the BT-RE subtask (6.2.1) are also used in the TT-RE (6.2.2) and TM-RE (6.2.3) subtasks.

Impact of Dataset Hyperparameters To determine the best configurations, we conduct an experiment to assess the impact of different combinations of dataset hyperparameters. These hyperparameters encompass the dataset weights, negative sample multiplier, and exclusion of articles with relation outliers. All RE classification models in this experiment are trained using `BioLinkBERT-base` as the pretrained model, and we evaluate them on the development dataset. Table 4 presents the results.

Lower negative sample multiplier values produce high recall but low precision, indicating a tendency to generate an excessive number of false positive relations. In contrast, higher negative sample multiplier values provide a better balance between precision and recall, resulting in the best overall performance.

We also observe that removing articles with relation outliers often improves performance. Finally, from the experiments, there does not emerge clear performance trends across different datasets and dataset weight combinations.

Hence, for the subsequent experiments and the final classification models, a negative sample multiplier of 10 is used, the articles with relation outliers are excluded, and all dataset and weight combinations are still considered.

RE Classification Model Performance We evaluate the performance of the seven pretrained models that are described in Section 4.3 with dense classification heads. We train the classification models on the platinum and gold datasets, with outlier articles excluded and a negative sample multiplier of 10. Table 5 shows that the results are similar across the different pretrained models. `BioLinkBERT-base` yields the best performance among the base models, while `BiomedBERT-large-uncased-abstract` and `BiomedElectra-large-uncased-abstract` yield the highest scores among the large models.

RE Ensemble Performance Similar to the NER ensemble experiment, the top-*n* RE systems from the previous RE experiments are used in a relation-level ensemble (Section 4.3.3). However, due to time constraints, we only consider ensemble sizes of 3 and 5. Table 6 shows the performance of relation-level ensembles of sizes 3 and 5 for each RE subtask. The level of performance is very similar, and it appears that the two extra RE systems in the ensemble of size 5 do not lead to an advantage.

5.5. GutBrainIE Submission Results

When preparing for the submission to GutBrainIE, we retrained the classification models of the best-performing systems from the previous experiments on a training dataset that included the development

Table 4
The F1_{micro} scores on the development dataset of RE classification models using `BioLinkBERT-base` under different experimental configurations with varying combinations of datasets, dataset weights, negative sample multiplier (NSM), and removal of articles with relation outliers. The entries in the "With Outliers" column for configurations using only platinum and gold datasets are omitted, as all outliers originate from the silver dataset.

w_p	w_g	w_s	w_b	NSM	With Outliers	Without Outliers
1	1	-	-	1	-	0.7505
1	1	1	-	1	0.7208	0.7186
1.5	1.5	1	0.75	1	0.7125	0.7391
1.25	1.25	1	0.75	1	0.7171	0.7305
1	1	-	-	3	-	0.7562
1	1	1	-	3	0.7345	0.7606
1.5	1.5	1	0.75	3	0.7612	0.7458
1.25	1.25	1	0.75	3	0.7603	0.7736
1	1	-	-	5	-	0.7645
1	1	1	-	5	0.7543	0.7621
1.5	1.5	1	0.75	5	0.7612	0.7732
1.25	1.25	1	0.75	5	0.7680	0.7757
1	1	-	-	10	-	0.7805
1	1	1	-	10	0.7642	0.7826
1.5	1.5	1	0.75	10	0.7708	0.7934
1.25	1.25	1	0.75	10	0.7711	0.7965

Table 5
The F1_{micro}, $\text{precision}_{\text{micro}}$, and $\text{recall}_{\text{micro}}$ scores on the development dataset of RE classification models that are trained on the platinum and gold datasets with a negative sample multiplier of 10 and the articles with relation outliers excluded. The pretrained models yielding the highest F1_{micro} scores are in bold.

Pretrained Model	$\text{Precision}_{\text{micro}}$	$\text{Recall}_{\text{micro}}$	F1_{micro}
`BioLinkBERT-base`	**0.7379**	**0.8318**	**0.7821**
`BioLinkBERT-large`	0.7358	0.8227	0.7768
`BiomedBERT-base-uncased-abstract`	0.7438	0.8182	0.7792
`BiomedBERT-base-uncased-abstract-fulltext`	0.7094	0.8545	0.7753
`BiomedBERT-large-uncased-abstract`	**0.7354**	**0.8591**	**0.7925**
`BiomedElectra-base-uncased-abstract`	0.7059	0.8727	0.7804
`BiomedElectra-large-uncased-abstract`	**0.7629**	**0.8045**	**0.7832**

dataset to make use of all the available annotated data. When all available data is used for training, no separate dataset remains for validation during training. Therefore, we train the classification models both with and without the development dataset to account for this limitation, in case those trained on the development dataset perform unexpectedly. When the development dataset is included in the training, it is assigned the same dataset weight as the platinum and gold datasets.

5.5.1. Named-Entity Recognition (Subtask 6.1)

For the NER subtask, we use the configurations presented in Table 2. We select the top 17 NER systems based on the results from all previous experiments. This selection is conducted separately for NER systems with the classification models trained with and without the development dataset included in the training data. Based on these selections, we create entity-level ensembles using the top-n NER systems, where $n \in \{3, 5, 7, 9, 11, 13, 15, 17\}$. This results in two sets of eight entity-level ensembles.

The test results for the NER subtask are shown in Table 7. As expected, including the development dataset in the training dataset increases the performance. The best result is achieved with an entity-level

Table 6
The F1_{micro}, $\text{precision}_{\text{micro}}$, and $\text{recall}_{\text{micro}}$ scores on the development dataset of relation-level ensembles of size 3 and 5 for each RE subtask.

Ensemble Size	$\textbf{Precision}_{\textbf{micro}}$	$\textbf{Recall}_{\textbf{micro}}$	$\textbf{F1}_{\textbf{micro}}$
BT-RE (Subtask 6.2.1)			
3	0.7530	0.8591	0.8025
5	0.7540	0.8500	0.7991
TT-RE (Subtask 6.2.2)			
3	0.7090	0.8261	0.7631
5	0.7209	0.8087	0.7623
TM-RE (Subtask 6.2.3)			
3	0.5489	0.6518	0.5959
5	0.5786	0.6375	0.6066

ensemble of the five NER systems with classification models trained with the development dataset included.

Table 7
The F1_{micro}, $\text{precision}_{\text{micro}}$, and $\text{recall}_{\text{micro}}$ scores on the test dataset for entity-level ensembles of varying sizes. Results are shown for NER systems with classification models trained with and without the development dataset. The result with the highest F1_{micro} is in bold.

Ensemble Size	$\textbf{Precision}_{\textbf{micro}}$	$\textbf{Recall}_{\textbf{micro}}$	$\textbf{F1}_{\textbf{micro}}$
Classification Models Trained with Development Dataset			
3	0.8181	0.8367	0.8273
5	**0.8286**	**0.8480**	**0.8382**
7	0.8218	0.8424	0.8319
9	0.8219	0.8432	0.8324
11	0.8255	0.8416	0.8335
13	0.8251	0.8407	0.8333
15	0.8303	0.8424	0.8363
17	0.8266	0.8399	0.8332
Classification Models Trained without Development Dataset			
3	0.8121	0.8351	0.8234
5	0.8191	0.8351	0.8271
7	0.8240	0.8367	0.8303
9	0.8288	0.8375	0.8331
11	0.8207	0.8327	0.8266
13	0.8243	0.8343	0.8292
15	0.8217	0.8310	0.8264
17	0.8225	0.8319	0.8272

5.5.2. Relation Extraction (Subtask 6.2)

Table 3 shows that the entity-level ensemble of size 9 achieves the best performance on the NER subtask without including the development dataset in the training dataset. Since this ensemble yields the highest score prior to evaluation on the test dataset, we select it as the best configuration. By extension, we also select the entity-level ensemble size of 9 trained with the development dataset, assuming it may yield the best performance. Hence, these two ensembles are used to generate the entities for the subsequent RE subtasks.

Based on the results shown in Table 5, the three pretrained models resulting in the best performance are `BioLinkBERT-base`, `BiomedBERT-large-uncased-abstract`, and `BiomedElectra-large-uncased-abstract`. Hence, the RE systems in all three RE subtasks use these models. Due to time constraints, the classification models with large pretrained models are trained only using the platinum and gold datasets, while those with the base model are trained on all four datasets and dataset weight combinations shown in Table 4.

For each RE subtask, we use the relation-level ensembles of size 3. This is due to time constraints, as well as what we observed in Table 6, where there is no clear difference between the relation-level ensembles of 3 or 5 RE systems.

Since all models for the submission are trained with and without the development dataset included, there are two entity-level ensembles and for each RE subtask there are two relation-level ensembles. Combined this results in four relation-level ensembles for each RE subtask.

Table 8 shows the test results for the RE subtasks. For each subtask, the best result is achieved by using the entities from the entity-level ensemble that is trained with the development dataset and by using the relation-level ensemble that is trained without the development dataset.

Table 8
The $F1_{micro}$, $precision_{micro}$, and $recall_{micro}$ scores on the test dataset for relation-level ensembles of size 3. Results are shown for classification models trained with and without the development (dev) dataset. The result for each RE subtask with the highest $F1_{micro}$ is in bold.

RE Trained with Dev	NER Trained with Dev	$\text{Precision}_{\text{micro}}$	$\text{Recall}_{\text{micro}}$	F1_{micro}
		BT-RE (Subtask 6.2.1)		
-	-	0.6450	0.7316	0.6856
✓	-	0.6350	0.7229	0.6761
-	✓	**0.6304**	**0.7532**	**0.6864**
✓	✓	0.6236	0.7316	0.6733
		TT-RE (Subtask 6.2.2)		
-	-	0.6329	0.7449	0.6843
✓	-	0.6178	0.7449	0.6754
-	✓	**0.6280**	**0.7572**	**0.6866**
✓	✓	0.6087	0.7490	0.6716
		TM-RE (Subtask 6.2.3)		
-	-	0.4137	0.5013	0.4533
✓	-	0.3893	0.4477	0.4165
-	✓	**0.4215**	**0.5148**	**0.4635**
✓	✓	0.3933	0.4598	0.4240

5.5.3. Final Leaderboards

Table 9 illustrates the final leaderboards for each of the subtasks of the top three teams in the GutBrainIE task. Our proposed information extraction pipeline achieved second place in the NER subtask (6.1), and first place in all three RE subtasks (6.2.1, 6.2.2, and 6.2.3).

In the NER subtask (6.1), the $F1_{micro}$ score difference between first place and our result is 0.0026, corresponding to a difference of 0.0031%. In the RE tasks, our approach outperformed second place by 0.0291, 0.0408, and 0.0906 for subtasks 6.2.1, 6.2.2, and 6.2.3, respectively. This corresponds to a difference in performance of 4.43%, 6.32%, and 24.3%, respectively.

Table 9
The final leaderboards for each of the subtasks of the top three teams in the GutBrainIE task. Only $F1_{micro}$ is presented, as it is the official evaluation metric of the GutBrainIE task.

	Subtask 6.1		Subtask 6.2.1		Subtask 6.2.2		Subtask 6.2.3	
Ranking	**Team**	$\mathbf{F1_{micro}}$	**Team**	$\mathbf{F1_{micro}}$	**Team**	$\mathbf{F1_{micro}}$	**Team**	$\mathbf{F1_{micro}}$
1	GutUZH	0.8408	**Gut-Instincts**	**0.6864**	**Gut-Instincts**	**0.6866**	**Gut-Instincts**	**0.4635**
2	**Gut-Instincts**	**0.8382**	ONTUG	0.6573	ataupd2425-pam	0.6458	Graphswise-1	0.3729
3	NLPatVCU	0.8370	Graphswise-1	0.6538	ONTUG	0.6443	ICUE	0.3651

6. Conclusion

In this paper, we proposed a biomedical information extraction pipeline, leveraging transformer-based models pretrained on biomedical corpora for NER and RE, as part of the GutBrainIE task.

To address the domain-specific terminology involved in the tasks, we relied on biomedical pretrained models, combined in ensembles. Specifically, for NER we selected and trained seven different pretrained models combined with three different classification heads. Based on individual model performance, we formed token-level and entity-level ensembles, ranging in size from 3 to 17. We discovered that both methods and all ensemble sizes led to significant improvements in performance and that entity-level ensembling consistently outperformed token-level ensembling. Performance across ensemble sizes was comparable, with only minor variations. For RE, we applied the same set of pretrained models as in NER and evaluated ensembles of sizes 3 and 5. Both ensemble sizes improved performance, with only minor differences observed between them.

Moreover, to address the availability of datasets of varying quality, we introduced a weighted training method, which improved performance for both NER and RE. Finally, for RE, we further improved the performance by introducing negative samples. Our experiments suggested that a large amount of negative samples (10 negative samples for each positive one) achieved the best performance.

The organizers of the GutBrainIE task conducted an external assessment of the performance of our information extraction pipeline. Our pipeline achieved the second-best performance in the NER subtask (6.1) and the best performance in the RE subtasks (6.2.1, 6.2.2, and 6.2.3).

For future work, we aim to improve the ability of the information extraction pipeline to distinguish between Chemical and Gene entities, as it currently struggles with it. This could be addressed by incorporating domain-specific gazetteers. For RE, we want to investigate the implementation of typed entity markers, as they may provide more contextual cues to improve relation classification. Furthermore, we want to improve data preparation by addressing the presence of HTML tags in the input texts. These could not be removed during the challenge, as the annotations of the test data were hidden. Removing HTML tags prevents the introduction of spurious tokens during tokenization, reducing the risk of the model learning irrelevant patterns and improving performance.

Acknowledgments

We gratefully acknowledge the support of CLAAUDIA through AI-Lab, which provided the computational infrastructure essential for conducting the experiments in this study.

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

During the preparation of this work, the authors used **Grammarly** and **Writefull** in order to: Grammar and spelling check, and Paraphrase and reword. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] I. Sekirov, S. L. Russell, L. C. M. Antunes, B. B. Finlay, Gut microbiota in health and disease, Physiological reviews (2010).

[2] E. A. Mayer, K. Tillisch, A. Gupta, Gut/brain axis and the microbiota, J. Clin. Invest. 125 (2015) 926–938.

[3] N. Perera, M. Dehmer, F. Emmert-Streib, Named Entity Recognition and Relation Detection for Biomedical Information Extraction, Front. Cell Dev. Biol. 8 (2020).

[4] E. W. Sayers, J. Beck, E. E. Bolton, J. R. Brister, J. Chan, R. Connor, M. Feldgarden, A. M. Fine, K. Funk, J. Hoffman, et al., Database resources of the National Center for Biotechnology Information in 2025, Nucleic Acids Research 53 (2024) D20.

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

[6] 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.

[7] L. A. Ramshaw, M. Marcus, Text Chunking using Transformation-Based Learning, in: VLC@ACL, 1995.

[8] A. Aid, R. Azzoune, I. Abdellaoui, A. Haddouche, Re-SciBERT: An Entity-Enriched Language Model to Enhance Biomedical Relation Extraction, in: 2024 1st International Conference on Innovative and Intelligent Information Technologies (IC3IT), IEEE, 2024, pp. 1–6.

[9] 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.

[10] 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, in: J. C. de Albornoz, J. Gonzalo, L. Plaza, A. G. S. de Herrera, J. Mothe, F. Piroi, P. Rosso, D. Spina, G. Faggioli, N. Ferro (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Sixteenth International Conference of the CLEF Association (CLEF 2025), 2025.

[11] N. Lawson, K. Eustice, M. Perkowitz, M. Yetisgen-Yildiz, Annotating large email datasets for named entity recognition with mechanical turk, in: Proceedings of the NAACL HLT 2010 workshop on creating speech and language data with Amazon's Mechanical Turk, 2010, pp. 71–79.

[12] R. Tinn, H. Cheng, Y. Gu, N. Usuyama, X. Liu, T. Naumann, J. Gao, H. Poon, Fine-tuning large neural language models for biomedical natural language processing, Patterns 4 (2023) 100729.

[13] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, in: NAACL-HLT (1), Association for Computational Linguistics, 2019, pp. 4171–4186.

[14] 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, Bioinformatics 36 (2019) 1234–1240.

[15] S. Alrowili, V. Shanker, BioM-Transformers: Building Large Biomedical Language Models with BERT, ALBERT and ELECTRA, in: D. Demner-Fushman, K. B. Cohen, S. Ananiadou, J. Tsujii (Eds.), Proceedings of the 20th Workshop on Biomedical Language Processing, Association for Computational Linguistics, 2021, pp. 221–227.

[16] K. Huang, J. Altosaar, R. Ranganath, ClinicalBERT: Modeling clinical notes and predicting hospital readmission, arXiv preprint arXiv:1904.05342 (2019).

[17] K. Huang, A. Singh, S. Chen, E. Moseley, C.-Y. Deng, N. George, C. Lindvall, Clinical XLNet: Modeling Sequential Clinical Notes and Predicting Prolonged Mechanical Ventilation, in: A. Rumshisky, K. Roberts, S. Bethard, T. Naumann (Eds.), Proceedings of the 3rd Clinical Natural Language Processing Workshop, Association for Computational Linguistics, 2020, pp. 94–100.

[18] K. Clark, M. Luong, Q. V. Le, C. D. Manning, ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators, in: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020, OpenReview.net, 2020.

[19] B. Wang, Q. Xie, J. Pei, Z. Chen, P. Tiwari, Z. Li, J. Fu, Pre-trained Language Models in Biomedical Domain: A Systematic Survey, ACM Comput. Surv. 56 (2023).

[20] J. Li, Y. Sun, R. J. Johnson, D. Sciaky, C.-H. Wei, R. Leaman, A. P. Davis, C. J. Mattingly, T. C. Wiegers, Z. Lu, BioCreative V CDR task corpus: a resource for chemical disease relation extraction, Database 2016 (2016) baw068.

[21] R. I. Doğan, R. Leaman, Z. Lu, NCBI disease corpus: A resource for disease name recognition and concept normalization, Journal of Biomedical Informatics 47 (2014) 1–10.

[22] L. Smith, L. K. Tanabe, R. J. N. Ando, C.-J. Kuo, I.-F. Chung, C.-N. Hsu, Y.-S. Lin, R. Klinger, C. M. Friedrich, K. Ganchev, et al., Overview of BioCreative II gene mention recognition, Genome biology 9 (2008) 1–19.

[23] M. Herrero-Zazo, I. Segura-Bedmar, P. Martínez, T. Declerck, The DDI corpus: An annotated corpus with pharmacological substances and drug–drug interactions, Journal of Biomedical Informatics 46 (2013) 914–920.

[24] À. Bravo, J. P. González, N. Queralt-Rosinach, M. Rautschka, L. I. Furlong, Extraction of relations between genes and diseases from text and large-scale data analysis: implications for translational research, BMC Bioinform. 16 (2015) 55:1–55:17.

[25] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, C. Dyer, Neural Architectures for Named Entity Recognition, in: K. Knight, A. Nenkova, O. Rambow (Eds.), Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, 2016, pp. 260–270.

[26] F. Souza, R. Nogueira, R. Lotufo, Portuguese named entity recognition using BERT-CRF, arXiv preprint arXiv:1909.10649 (2019).

[27] T. Almeida, R. A. Jonker, R. Poudel, J. M. Silva, S. Matos, BIT.UA at MedProcNer: Discovering Medical Procedures in Spanish Using Transformer Models with MCRF and Augmentation., in: CLEF (Working Notes), 2023, pp. 60–72.

[28] Z. Dai, X. Wang, P. Ni, Y. Li, G. Li, X. Bai, Named Entity Recognition Using BERT BiLSTM CRF for Chinese Electronic Health Records, in: 2019 12th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), 2019, pp. 1–5.

[29] L. Baldini Soares, N. FitzGerald, J. Ling, T. Kwiatkowski, Matching the Blanks: Distributional Similarity for Relation Learning, in: A. Korhonen, D. Traum, L. Màrquez (Eds.), Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, 2019, pp. 2895–2905.

[30] K. He, R. Mao, T. Gong, E. Cambria, C. Li, JCBIE: a joint continual learning neural network for biomedical information extraction, BMC bioinformatics 23 (2022) 549.

[31] J. Zhu, J. Dong, H. Du, Y. Geng, S. Fan, H. Yu, Z. Shao, X. Wang, Y. Yang, W. Xu, Tell me your position: Distantly supervised biomedical entity relation extraction using entity position marker, Neural Networks 168 (2023) 531–538.

[32] W. Yoon, S. Yi, R. Jackson, H. Kim, S. Kim, J. Kang, Biomedical relation extraction with knowledge base–refined weak supervision, Database 2023 (2023) baad054.

[33] X. Liu, J. Tan, J. Fan, K. Tan, J. Hu, S. Dong, A Syntax-enhanced model based on category keywords for biomedical relation extraction, Journal of Biomedical Informatics 132 (2022) 104135.

[34] N. Bölücü, M. Rybinski, X. Dai, S. Wan, An adaptive approach to noisy annotations in scientific information extraction, Information Processing & Management 61 (2024) 103857.

[35] Z. Wang, J. Shang, L. Liu, L. Lu, J. Liu, J. Han, CrossWeigh: Training Named Entity Tagger from Imperfect Annotations, in: K. Inui, J. Jiang, V. Ng, X. Wan (Eds.), Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, 2019, pp. 5154–5163.

[36] Y. Meng, Y. Zhang, J. Huang, X. Wang, Y. Zhang, H. Ji, J. Han, Distantly-Supervised Named

Entity Recognition with Noise-Robust Learning and Language Model Augmented Self-Training, in: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2021, pp. 10367–10378.

[37] D. T. Nguyen, C. K. Mummadi, T. P. N. Ngo, T. H. P. Nguyen, L. Beggel, T. Brox, Self: Learning to filter noisy labels with self-ensembling, arXiv preprint arXiv:1910.01842 (2019).

[38] I. Loshchilov, F. Hutter, Decoupled Weight Decay Regularization, in: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019, OpenReview.net, 2019.

[39] D. S. Kalra, M. Barkeshli, Why Warmup the Learning Rate? Underlying Mechanisms and Improvements, in: The Thirty-eighth Annual Conference on Neural Information Processing Systems, 2024.