Comparing CRF vs BERT Models for Named Entity Recognition and Relation Extraction

Lorenzo Pamio[1], Giorgio Maria Di Nunzio[1]

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

Abstract

This paper presents our participation in the CLEF 2025 GutBrainIE challenge, addressing tasks in Named Entity Recognition (NER) and Relation Extraction (RE) on biomedical texts related to the gut-brain axis. We explored both traditional and modern approaches, including Conditional Random Fields (CRFs) with hand-engineered features and fine-tuned BERT-based models. For RE, we focused on a simplified pipeline using *BiomedBERT*, coupled with NER outputs to extract binary and ternary relations. Our experiments revealed the limitations of CRFs in this domain and highlighted the variability and sensitivity of BERT-based models to training stability and dataset noise. While our NER performance was mid-ranked, we achieved competitive results in RE, particularly in ternary tag-based extraction. We also reflect on the effects of model selection, loss function design, and data configurations, offering insights for future work in biomedical IE.

Keywords

CRF model, BERT model, Fine tuning, NER, RE

1. Introduction

CLEF[1] (Conference and Labs of the Evaluation Forum) is an annual initiative that hosts a series of challenges and tasks focused on information access systems. These challenges cover a broad spectrum, including evaluation methodologies, metrics, and the presentation of new data collections. This paper focuses on the GutBrainIE tasks within the BioASQ laboratory 2025 [1] , specifically subtask 6.1 on Named Entity Recognition (NER), and subtasks 6.2.1 (Binary Relation Extraction), 6.2.2 (Ternary Tag-based Relation Extraction), and 6.2.3 (Ternary Mention-based Relation Extraction), all of which are about Relation Extraction (RE). The domain of these challenges centers on biomedical literature, with a particular emphasis on the gut-brain axis and its relation to neurological diseases [2].

Our approach to each subtask varies, combining traditional models such as Conditional Random Fields with fine-tuned BERT-based models. The main objective is to develop reliable models with strong performance on the domain-specific data, aiming for top rankings in the final evaluation.

The paper is organized as follows: Section 2 introduces related works; Section 3 describes our approach; Section 4 explains our experimental setup; Section 5 discusses our main findings; finally, Section 6 draws some conclusions and outlooks for future work.

2. Related Work

Named Entity Recognition is a task of Natural Language Processing that has as objective classifying entities inside text, we will refer to this type of task with the acronym of NER. Early approaches to solve this task were using rule-based systems and feature engineering often using models like Conditional Random Fields (CRFs) [3]. With the rise of deep learning, neural network architectures have become dominant. More specifically, transformer-based models like BERT [4] have really improved the performance in this task. Relation extraction similarly focuses on identifying relationships between

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

lorenzo.pamio@studenti.unipd.it (L. Pamio); giorgiomaria.dinunzio@unipd.it (G. Di Nunzio)

https://www.dei.unipd.it/~dinunzio/ (G. Di Nunzio)

0009-0003-6201-095X (L. Pamio); 0000-0001-9709-6392 (G. Di Nunzio)

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

entities, during this paper we will refer to this tasks with the acronym of RE. Recent advancements in deep learning have significantly improved performance in this task as well. In particular, models such as *BiomedBERT* [5], developed by Microsoft, have contributed substantially to progress in the biomedical domain. The work presented in this paper builds upon the foundation established by *BiomedBERT*, which serves as a core component of our approach.

3. Methodology

We will now define briefly the subtasks about NER and RE, to get more about

3.1. Named Entity Recognition

Given a set L of *labels*, an ordered sequence T of *tokens* of size n and a set R of functions that can map a token into a label, we formally define the problem of NER as:

$$r_i : T^n \to L^n, \quad \text{where} \quad r_i((t_1, ..., t_n)) = (l_1, ..., l_n), r_i \in R \tag{1}$$

The objective of this task is to assign a label to each token in a given token set, minimizing the overall *loss* with respect to a known ground truth. This process should ideally consider not only individual tokens but the entire token sequence T for context-aware predictions.

$$\begin{aligned} \mathcal{L}(r_i) &= \sum_j \text{loss}(r_i(T^{(j)}), L^{(j)}) \\ r^* &= \underset{r_i \in R}{\text{argmin}} \quad \mathcal{L}(r_i) \end{aligned} \tag{2}$$

The *loss* can be defined in various ways. Ideally, it could be expressed as the negative of a reward function, allowing us to optimize for the function that yields the best overall performance.

The ideal approach to the task assumes that the loss can be computed efficiently for a given label set L. However, in real-world scenarios, this *loss* function is often not directly computable due to the inherent ambiguity in assigning a token to a specific label, as well as the subjective nature of human annotation that may label the same entity differently. In practice, applying this approach requires defining the initial token set T as a sequence of tokens that, when combined, reconstruct the document. Similarly, the label domain L is constrained by the task's scope, and the number of labels is limited to a finite, positive integer.

3.2. Relation Extraction

The RE task works given a set of entities E, with their attributes *text span*, *position* and *label*, and a set of predicates P, the objective of the task is to associate possible relations between entities.

$$p \in P \quad e_1, e_2 \in E \quad l_1, l_2 \in L \tag{3}$$

The labels l_1, l_2 are defined as the labels associated with the entities e_1, e_2 respectively. The task can be specified in different ways depending on the subtask. In subtask 6.2.1, the objective is to determine whether a relation exists between the two given labels. Subtask 6.2.2 extends this requirement by also identifying the specific predicate that characterizes the relation. Subtask 6.2.3 further requires the extraction of the text spans corresponding to the related entities, in addition to identifying the predicate.

In the following formula Ob_i will refer to subtask 6.2.1 about binary tag-based RE, Ot_i will refer to subtask 6.2.2 about ternary tag-based RE and finally Otm_i will refer to subtask 6.2.3 about ternary mention-based RE.

$$\begin{aligned} Ob_i &= \{(l_1, l_2) | l_1, l_2 \in L, \text{if exists a relation between } l_1, l_2\} \\ Ot_i &= \{(l_1, p, l_2) | l_1, l_2 \in L, p \in P, \text{if exists a relation between } l_1, l_2\} \\ Otm_i &= \{(l_1, p, l_2, s_1, s_2) | l_1, l_2 \in L, p \in P, \text{if exists a relation between } l_1, l_2\} \end{aligned} \tag{4}$$

In the equation defining Otm_i the spans s_1, s_2 are defined as the spans in the text associated with respect to the labels and entities l_1, l_2

3.3. CRF model

We began by developing a model based on Conditional Random Fields (CRFs), aiming to build it from scratch and evaluate its performance in the specific domain of the GutBrainIE task. CRF models are statistical modeling methods that incorporate contextual information, making them well-suited for sequence labeling tasks like NER [3]. CRFs rely heavily on feature design and transition probabilities. Since the predictions are derived from input features, feature engineering plays a crucial role in determining the model's capabilities [3]. For this challenge, we designed a custom feature set tailored to the biomedical domain and the structure of the provided texts.

The CRF model was modified in different ways from the default configuration and its hyperparameters has been tweaked to obtain different types of performances. Specifically trained models were based on the package *sklearn_crfsuite*[2] which provides different training algorithms like *lbfgs* [6], *l2sgd* [7], *ap* [8], *pa* [9], *arow*[10]. Among these algorithms, the best performances have been obtained with the *lbfgs* method, which has therefore been chosen for being integrated in the final model.

In addition to the training algorithm, several important parameters were tuned to control the model's regularization behavior and feature handling:

- *c1*, value responsible for the LASSO regression [11]
- *c2*, value responsible for the RIDGE regression [12]
- *all_possible_transitions*, boolean value responsible for evaluating even the non-present transition in the training dataset
- *min_freq*, value responsible for evaluating the minimal frequency in which a feature needs to be present to be taken into account by the model

The feature engineering applied to these CRF models involves a standard set of features used to label tokens and extract relations. The core idea behind feature engineering is to process an entire document token by token, extracting specific features for each token as well as information about its surrounding context [3]. In the specific case of this challenge, to represent the current token, we used the following information:

1. '*word.lower()*': the lowercase representation of the token
2. '*word[-3:]*': last 3 chars of the token
3. '*word[-2:]*': last 2 chars of the token
4. '*word.isupper()*': if the token is uppercase
5. '*word.istitle()*': if the token is title
6. '*word.hasCapital()*': if the token has capital letter
7. '*word.isdigit()*': if the token is a digit
8. '*word.isGene()*': a custom implementation, if the token is a scientific representation of a gene
9. '*postag*': postag of token
10. '*postag[:2]*': first 2 chars of postag
11. '*word.length()*': length of token
12. '*word.pos()*': postion of the token in the phrase

We also incorporated, whenever possible, features derived from the preceding and following tokens to enrich the representation of the current token. These contextual features consist of a subset of those used for the current token itself, specifically features 1, 4, 5, 9, and 10, i.e., *word.lower*, *word.isupper*, *word.istitle*, *postag*, and *postag[:2]*.

[2] https://sklearn-crfsuite.readthedocs.io/en/latest/

3.4. BERT models

In addition to CRF models, we also adopted an approach based on fine-tuning pre-trained models. This fine-tuning process aimed to improve the base performance of various models available through the HuggingFace library[3]. Several types of models were considered, and each was specifically trained to achieve the best possible performance within the subtasks constraints. The models we fine-tuned and subsequently submitted to the challenge were:

- *scibert-scivocab-uncased*[4]
- *biobert-base-cased-v1.2*[5]
- *BiomedNLP-BiomedBERT-base-uncased-abstract*[6]
- *biosyn-sapbert-bc2gn*[7]
- *NuNER-v2.0*[8]

All of them (except *NuNER-v2.0*) were specifically pre-trained on scientific and/or bio-related corpora of documents that enhanced the performance in our specific domain.

4. Experimental Setup

4.1. Dataset

The datasets provided by the competition organizers are composed of entities and relationships between them inside titles and abstracts of PubMed abstracts.

Regarding the challenge, the provided datasets include:

- Entity Mentions: Text spans classified into predefined categories.
- Relations: Associations between entities, specifying that a particular relationship holds between two entities.

In the specific instance of the GutBrainIE challenge, the corpus of documents was annotated in different ways:

- Platinum collection, highest-quality annotations, expert-curated and reviewed by external biomedical specialists.
- Gold collection, high-quality annotations, expert-curated.
- Silver collection, mid-quality annotations, created by trained students under expert supervision.
- Bronze collection, automatically generated annotations.
- Dev collection, used as test set.

Working on the 6.1 subtask about NER, our setup was split into two main working pipeline regarding respectively a CRF model (Section 3.3) trained from scratch and a pipeline to fine tune BERT models (Section 3.4).

4.2. Named Entity Recognition

The setup used in this subtask is mainly related to the hyperparameters of the models themselves. We also tweaked the domain and format of the training set used for the task, although the main focus in this part of the challenge was placed more on the models than on data processing.

[3] https://huggingface.co/docs/huggingface_hub/guides/overview
[4] https://huggingface.co/allenai/scibert_scivocab_uncased
[5] https://huggingface.co/dmis-lab/biobert-base-cased-v1.2
[6] https://huggingface.co/microsoft/BiomedNLP-BiomedBERT-base-uncased-abstract
[7] https://huggingface.co/dmis-lab/biosyn-sapbert-bc2gn
[8] https://huggingface.co/numind/NuNER-v2.0

Table 1
Parameters settings for submitted CRF models

model	c1	c2	dataset	min_freq
customCRF-OVER	$1 \cdot 10^{-6}$	$1 \cdot 10^{-6}$	Complete	0
customCRF-default	0.1	0.1	Complete	0
customCRF-LowF-noDevBronze	$1 \cdot 10^{-6}$	$1 \cdot 10^{-6}$	No Dev,Bronze	0

Table 2
Parameters settings for submitted BERT fine tuned models

model	`warmup_ratio`	`weight_decay`	`learning_rate`
scibert-scivocab-uncased	0.06	0.01	$5e^{-5}$
biobert-base-cased-v1.2	0.06	0.05	$1e^{-5}$
BiomedNLP-BiomedBERT-base-uncased-abstract	0.1	0.05	$1e^{-5}$
biosyn-sapbert-bc2gn	0.1	0.01	$1e^{-5}$
NuNER-v2.0	0.1	0.05	$5e^{-5}$

Table 3
Micro-scores for the submitted NER models, tested in a never-seen dataset (Dev)

	micro		
Model	Precision	Recall	F1-Score
BiomedBERT	0.5130	0.7896	0.6220
biosyn-sapbert-bc2gn-8	0.5392	0.8173	0.6498
biosyn-sapbert-bc2gn-12	0.5649	0.7788	0.6548
NuNER-v2.0-22-CW	0.6257	0.8128	0.7071
scibert-27	0.4988	0.7797	0.6084
scibert-47	0.5514	0.7681	0.6419
customCRF-OVER	0.5601	0.4172	0.4782
customCRF-default	0.4091	0.4691	0.4370

4.2.1. CRF models

The main differences between the setup and the overall models produced with CRF are shown in Table 2. We mainly adjusted values associated with regularization functions, specifically L1 (`c1_value`) and L2 (`c2_value`). The `min_freq` parameter was kept at 0 to ensure that every feature present in the training dataset was captured. We also varied the amount and type of data used for training.

4.2.2. BERT models

BERT (Bidirectional Encoder Representations from Transformers) is a transformer-based model pre-trained on large corpora using a masked language modeling objective [4]. Its success in a wide range of NLP tasks has made it a natural choice for sequence classification and token-level prediction tasks. A key concern with BERT models and the training pipeline was the stability of the process. Indeed, in some training iterations, the loss function fluctuated significantly, leading to considerable variation in the results. To address this and improve stability, we adjusted the unstable models' hyperparameters.

We also decided to use only one model implementing the CustomWeight loss function, as most of the domain-specific scientific or biomedical models did not yield the performance improvements we had hoped for.

4.3. Relation Extraction

For the RE subtasks, we relied heavily on a single model; the *BiomedNLP-BiomedBERT-base-uncased-abstract* model, focusing more on optimizing one single model for all the RE subtasks. To extract relations from the text, the RE model had to be paired with a NER model capable of identifying the entities to be used in the subsequent steps.

We decided to experiment with the following fine-tuned NER models[9]:

- *biosyn-sapbert-bc2gn-12*[10]
- *scibert-27*[11]
- *NuNerv2.0-22-CW-xtreme*[12]

The *biosyn-sapbert-bc2gn-12* model has been chosen because it was expected to have the best theoretical performance due to its scientific and biorelated pre-training.

The *scibert-27* model has been chosen because the 47-epoch version seemed like a model that could have overfitted over some of the data.

The *NuNerv2.0-22-CW-xtreme* model has been chosen because it had the most generic domain training background, it had the best performance in unseen data and because it was relying on our CustomWeight loss function.

During the development of these RE models, we defined a metric that was used as the main varying parameter. This parameter has been called *norel_ratio*.

$$norel\_ratio = \frac{|N|}{|P|} \tag{5}$$

Where N is a set of relations that are labeled as negative, denoting a non-existing link between two entities in the text. The P set consists of existing relation between entities in the text. In the specific instance of this study, we always used the entirety of the positive instances of relation as a starting point to compute the N set of non-existing relations. To create the set N of negative instances we have used a random approach, extracting and inserting in this set relationships that didn't exist between random entities. These models actually have been trained with 3 iterations of the *BiomedBERT* RE model, where the *norel_ratio* has been tweaked and ranged from 1 to 3

5. Results

The total number of submitted runs was 37. Out of these 37, 10 were related to the first subtasks about NER (3.1), and the remaining 27 were distributed equally over the 3 RE (3.2) subtasks. As shown in table 5, the results on the NER subtask 6.1 show that BERT models had the best performance overall (considering the micro-f1 score as the reference metric).

The customCRF models, trained from scratch (see Section 3.3), did not perform as well as the other approaches. Similarly, the Custom Weight scheme, which was applied to the BERT models through a custom loss function and initially showed promising results during early evaluation, ultimately ranked lower both in terms of average position and Micro F1-score when compared to other BERT-based models. This result was expected, as we anticipated that the most general-purpose configuration would yield the weakest performance among the BERT variants.

Concerning RE (3.2) subtasks, average performances of proposed models are similar. Analyzing models' behaviors reported in Tables 6,7,8, we can see that overall the best micro-f1 score has been obtained with models having a higher ratio of no_relation over effective relation in the training dataset.

Even though the overall F1-score distribution was variable, it is worth noting that, in Task 6.2.2, some models trained with a ratio of 1 achieved a high macro-F1 score. This indicates strong performance across all relation classes, suggesting that these models were effective in distinguishing between different types of relations.

[9]These models have been fine-tuned in the NER subtask
[10]Base model at https://huggingface.co/dmis-lab/biosyn-sapbert-bc2gn
[11]Base model at https://huggingface.co/allenai/scibert_scivocab_uncased
[12]Base model at https://huggingface.co/numind/NuNER-v2.0

Table 4
Raw performances of RE models, the entities in this table were already extracted from the ground-truth files and used as a starting base for the RE model, effectively evaluating only the ability to extract relation of the models. The numbers x and y in the name of the model following the xe-yrel format have a specific meaning, x stand for the total epochs of training, y for the norel_ratio

		macro			micro		
model	Subtask	precision	recall	F1	precision	recall	F1
5e-2rel	6.2.1	0.5829	0.8194	0.6429	0.5856	0.8864	0.7052
5e-2rel	6.2.2	0.7330	0.7482	0.7097	0.7510	0.8130	0.7808
5e-2rel	6.2.3	0.5542	0.6838	0.5738	0.3960	0.7375	0.5153
3e-3rel	6.2.1	0.5927	0.5329	0.5245	0.6923	0.6545	0.6729
3e-3rel	6.2.2	0.5205	0.3962	0.4297	0.8613	0.5130	0.6431
3e-3rel	6.2.3	0.4377	0.3600	0.3950	0.4728	0.4804	0.4765
3e-1rel	6.2.1	0.4869	0.5534	0.5180	0.4826	0.9085	0.6303
3e-1rel	6.2.2	0.4423	0.4268	0.4344	0.5895	0.8231	0.6869
3e-1rel	6.2.3	0.3946	0.3742	0.3841	0.2674	0.6912	0.3856

Table 5
Results of submitted runs on subtask 6.1. Here the results point out that the overall best performances were obtained by fine-tuned BERT models, more specifically the *biosyn-sapbert-bc2gn* model was the best for the task of NER

	macro			micro		
model	P	R	F1	P	R	F1
biosyn-sapbert-bc2gn-12	0.6400	0.7435	0.6763	0.6809	0.7745	0.7247
biosyn-sapbert-bc2gn-8	0.6447	0.7383	0.6856	0.6778	0.7736	0.7225
scibert-47	0.6554	0.6987	0.6406	0.6736	0.7607	0.7145
scibert-27	0.6350	0.6997	0.6256	0.6641	0.7607	0.7091
BiomedNLP-BiomedBERT	0.6097	0.7079	0.6391	0.6571	0.7623	0.7058
biobert-base-cased-v1.2-14-CW-xtreme	0.6468	0.6804	0.6259	0.6720	0.7421	0.7053
NuNerv2.0-22-CW-xtreme	0.6256	0.7052	0.6186	0.6564	0.7567	0.7030
customCRF-LowF-noDevBronze	0.3605	0.3566	0.3470	0.4917	0.4527	0.4714
customCRF	0.4147	0.3380	0.3472	0.4098	0.4390	0.4239
customCRF-LowF	0.3790	0.2997	0.3174	0.4935	0.3670	0.4210

Table 6
Results of submitted runs on subtask 6.2.1. The results suggest that overall a higher values of *norel* ratios, will raise micro-f1 scores of the models, both because the models had more data to train with and because the model had more context to understand what was a real relation and what was not.

	macro			micro		
model	P	R	F1	P	R	F1
RE-BiomedNLP-3NoRel-1epoch	0.4807	0.6091	0.4993	0.5671	0.7316	0.6389
RE-BiomedNLP-3NoRel-1epoch	0.5020	0.5929	0.5003	0.5619	0.7273	0.6340
RE-BiomedNLP-3NoRel-1epoch	0.4682	0.6066	0.4952	0.5567	0.7229	0.6290
RE-BiomedNLP-2NoRel-1epoch	0.4338	0.6065	0.4741	0.5463	0.7403	0.6287
RE-BiomedNLP-2NoRel-1epoch	0.4528	0.6209	0.4805	0.5449	0.7359	0.6262
RE-BiomedNLP-2NoRel-1epoch	0.4386	0.6078	0.4741	0.5414	0.7359	0.6239
RE-BiomedNLP-1NoRel-1epoch	0.4580	0.7118	0.5176	0.5068	0.8095	0.6233
RE-BiomedNLP-1NoRel-1epoch	0.4908	0.7080	0.5272	0.5082	0.8052	0.6231
RE-BiomedNLP-1NoRel-1epoch	0.4413	0.7187	0.5165	0.4987	0.8095	0.6172

6. Conclusions and Future Work

Our participation in this task showed that for the NER subtasks, although we explored different approaches, our results were not among the top performers. However, the trend was different for the RE subtasks. We achieved satisfying results in subtask 6.2.2, and overall, our performances in the 6.2

Table 7
Results of submitted runs on subtask 6.2.2. These results actually point out the less coherent results in relation with the value of *norel* ratio. This behavior is still unknown but it could be that in this specific task the impact of the NER model used previously to the RE model was more important than in the other RE subtasks.

	macro			micro		
model	P	R	F1	P	R	F1
RE-BiomedNLP-3NoRel-1epoch	0.4409	0.5704	0.4694	0.5853	0.7202	0.6458
RE-BiomedNLP-2NoRel-1epoch	0.4398	0.6052	0.4812	0.5701	0.7366	0.6427
RE-BiomedNLP-2NoRel-1epoch	0.4325	0.6064	0.4787	0.5651	0.7325	0.6380
RE-BiomedNLP-3NoRel-1epoch	0.4554	0.5680	0.4729	0.5767	0.7119	0.6372
RE-BiomedNLP-1NoRel-1epoch	0.4411	0.6805	0.5003	0.5257	0.7984	0.6340
RE-BiomedNLP-3NoRel-1epoch	0.4278	0.5679	0.4638	0.5710	0.7119	0.6337
RE-BiomedNLP-1NoRel-1epoch	0.4290	0.6870	0.5017	0.5229	0.7984	0.6319
RE-BiomedNLP-2NoRel-1epoch	0.4427	0.6097	0.4776	0.5570	0.7243	0.6297
RE-BiomedNLP-1NoRel-1epoch	0.4419	0.6742	0.4980	0.5219	0.7860	0.6273

Table 8
Results of submitted runs on subtask 6.2.3. In this table the results are clear and they point out the fact that the best models are the ones that had more data to train and more context very similar as seen in the Table 6

	macro			micro		
model	P	R	F1	P	R	F1
RE-BiomedNLP-3NoRel-1epoch	0.1940	0.2764	0.1982	0.2278	0.3432	0.2738
RE-BiomedNLP-3NoRel-1epoch	0.2012	0.2872	0.2069	0.2270	0.3378	0.2716
RE-BiomedNLP-3NoRel-1epoch	0.1873	0.2718	0.1936	0.2179	0.3365	0.2645
RE-BiomedNLP-2NoRel-1epoch	0.1798	0.3063	0.1993	0.2064	0.3539	0.2607
RE-BiomedNLP-2NoRel-1epoch	0.1796	0.3142	0.2020	0.2080	0.3472	0.2602
RE-BiomedNLP-2NoRel-1epoch	0.1755	0.3021	0.1955	0.2014	0.3485	0.2553
RE-BiomedNLP-1NoRel-1epoch	0.1546	0.3223	0.1857	0.1786	0.3887	0.2447
RE-BiomedNLP-1NoRel-1epoch	0.1538	0.3124	0.1802	0.1766	0.3941	0.2439
RE-BiomedNLP-1NoRel-1epoch	0.1454	0.3050	0.1746	0.1734	0.3874	0.2395

Table 9
Overall results on the GutBrainIE test dataset, the Position column is associated with the results obtained overall among all the participants of the task, the *f1 diff with 1st place* column explains how much the micro-f1 scores were off the 1^{st} place micro-f1 scores

			macro			micro			
Subtask	Position	id	P	R	F1	P	R	F1	f1 diff with 1^{st} place
6.1	13/16	3	0.6400	0.7435	0.6763	0.6809	0.7745	0.7247	0.1161
6.2.1	5/12	A7	0.4807	0.6091	0.4993	0.5671	0.7316	0.6389	0.0475
6.2.2	2/13	B7	0.4409	0.5704	0.4694	0.5853	0.7202	0.6458	0.0407
6.2.3	8/13	C7	0.1940	0.2764	0.1982	0.2278	0.3432	0.2738	0.1897

subtasks were better than in subtask 6.1, this results are summarized in Table 9.

Promising directions for future work include the evaluation of larger models performance gains they may bring in this specific domain. Additionally, we aim to investigate the optimal no_rel ratio and how changes to this parameter affect model performance, clarifying whether this value has a generally applicable threshold or if it is domain-dependent. In addition, we aim to integrate a semantic perspective grounded in linguistic analysis to enrich the linguistic and conceptual interpretation of extracted terms and relations. Specifically, we would like to apply semic analysis, which decomposes terms into minimal semantic units, as a structured approach to uncovering the internal organization of meaning in medical terminology [13, 14]. Incorporating this technique may enhance our ability to align terminological outputs with underlying conceptual structures, improving not only model interpretability but also the precision of the extraction of named entities and objects in domain-specific biomedical contexts.

Acknowledgments

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

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

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

[2] 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.

[3] J. D. Lafferty, A. McCallum, F. C. N. Pereira, Conditional random fields: Probabilistic models for segmenting and labeling sequence data, in: Proceedings of the Eighteenth International Conference on Machine Learning, ICML '01, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2001, pp. 282–289.

[4] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, CoRR abs/1810.04805 (2018). URL: http://arxiv.org/abs/1810.04805. arXiv:1810.04805.

[5] Y. Gu, R. Tinn, H. Cheng, M. Lucas, N. Usuyama, X. Liu, T. Naumann, J. Gao, H. Poon, Domain-specific language model pretraining for biomedical natural language processing, CoRR abs/2007.15779 (2020). URL: https://arxiv.org/abs/2007.15779. arXiv:2007.15779.

[6] D. C. Liu, J. Nocedal, On the limited memory bfgs method for large scale optimization, Mathematical programming 45 (1989) 503–528.

[7] L. Bottou, Stochastic Gradient Descent Tricks, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 421–436. URL: https://doi.org/10.1007/978-3-642-35289-8_25. doi:10.1007/978-3-642-35289-8_25.

[8] M. Collins, Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms, in: Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002), Association for Computational Linguistics, 2002, pp. 1–8. URL: https://aclanthology.org/W02-1001/. doi:10.3115/1118693.1118694.

[9] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, Y. Singer, Online passive-aggressive algorithms, Journal of Machine Learning Research 7 (2006) 551–585. URL: http://jmlr.org/papers/v7/crammer06a.html.

[10] K. Crammer, A. Kulesza, M. Dredze, Adaptive regularization of weight vectors, in: Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, A. Culotta (Eds.), Advances in Neural Information Processing Systems, volume 22, Curran Associates, Inc., 2009. URL: https://proceedings.neurips.cc/paper_files/paper/2009/file/8ebda540cbcc4d7336496819a46a1b68-Paper.pdf.

[11] R. Tibshirani, Regression shrinkage and selection via the lasso, Journal of the Royal Statistical Society. Series B (Methodological) 58 (1996) 267–288. URL: http://www.jstor.org/stable/2346178.

[12] A. E. Hoerl, R. W. Kennard, Ridge regression: Biased estimation for nonorthogonal problems, Technometrics 42 (2000) 80–86. URL: http://www.jstor.org/stable/1271436.

[13] V. Bonato, G. M. Di Nunzio, F. Vezzani, A Novel Approach to Semic Analysis: Extraction of

Atoms of Meaning to Study Polysemy and Polyreferentiality, Languages 9 (2024) 121. URL: https://www.mdpi.com/2226-471X/9/4/121. doi:10.3390/languages9040121, number: 4 Publisher: Multidisciplinary Digital Publishing Institute.

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