Multilingual Embedding and Prompt-Driven Approaches for Named Entity Recognition, Entity Linking, and Clinical Code Prediction in Greek Discharge Summaries

Notebook for the BioASQ Lab, CLEF 2025

Poojan Vachharajani[1]

[1]*Netaji Subhas University of Technology, New Delhi, India*

Abstract

This paper describes our participation in the ELCardioCC shared task under the team name pjmathematician, focusing on Named Entity Recognition (NER), Entity Linking (EL), and Multi-Label Classification with Explainable AI (MLC-X) over Greek cardiology discharge letters. For the NER and EL subtasks, we employed various configurations based on large language models, including Qwen2.5-32B-Instruct and a fine-tuned LoRA variant, combined with prompt engineering strategies. Entities were extracted from unstructured Greek clinical text and semantically matched to ICD-10 codes using embeddings from the multilingual-e5-large-instruct model. In the MLC-X task, we leveraged a larger Qwen2.5-72B model, guided by a candidate list of codes generated via multilingual document embeddings. This system explored cross-lingual semantic similarity and prompt-tuned LLMs to perform entity-level and document-level clinical coding in a low-resource language setting.

Keywords

Clinical Natural Language Processing, Named Entity Recognition, Entity Linking, Large Language Models (LLMs), ICD-10 Coding, Greek NLP

CEUR-WS.org/Vol-4038/paper_47.pdf

1. Introduction

Clinical Natural Language Processing (NLP) plays a crucial role in extracting valuable information from unstructured medical texts, such as discharge summaries. The ELCardioCC shared task [1] [8] aims to advance clinical coding in cardiology by focusing on Greek, a relatively low-resource language in the biomedical domain. The task is divided into three subtasks: Named Entity Recognition (NER), Entity Linking (EL) to ICD-10 codes, and Multi-Label Classification with Explainable AI (MLC-X) for assigning ICD-10 codes to entire documents.

Our team, pjmathematician, participated in all three subtasks. We leveraged Large Language Models (LLMs) for entity extraction and initial code prediction, combined with multilingual sentence embeddings for robust semantic matching to ICD-10 codes. This paper details our methodologies, experimental setups, and the results achieved for each subtask.

2. Dataset

The ELCardioCC task provided a specialized corpus of 1,000 Greek discharge letters from a cardiology department for training and validation, and 500 letters for testing. These documents were annotated with mentions (chief complaint, diagnosis, prior medical history, drugs, and cardiac echo) and their corresponding ICD-10 codes. The dataset reflects the complexities of real-world clinical narratives, including specialized terminology and abbreviations [2]. A separate ‘codes.csv‘ file containing ICD-10 codes and their descriptions, and a ‘labelset.txt‘ with the target ICD-10 codes for the task were also provided. For our fine-tuning experiments, the 1000-instance training set was split into a 700-instance training and 300-instance validation set.

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

pjmathematician@gmail.com (P. Vachharajani)

© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

3. Methods

Our approach varied across the subtasks, primarily utilizing LLMs for NER and initial predictions, and sentence embeddings for EL and supporting MLC-X.

3.1. Named Entity Recognition (NER) - Subtask 1

For NER, we experimented with three main configurations:

- **Config 1 & 2 (Base LLM)**: These two configurations were identical experimental runs to assess the stochasticity of the model’s output. Both used the base ‘Qwen/Qwen2.5-32B-Instruct‘ model [3]. Inference was performed using LMDeploy [4] with a ‘TurbomindEngineConfig‘ and a generative configuration set for sampling (‘top_p=0.8‘, ‘temperature=0.8‘). A detailed, zero-shot prompt (see Appendix A.1) was designed to instruct the LLM to translate the Greek text, detect entities, provide an English translation for each, and explain its relevance in a structured JSON format.
- **Config 3 (LoRA Fine-tuned LLM)**: This configuration utilized a LoRA [5] fine-tuned version of ‘Qwen/Qwen2.5-32B-Instruct-AWQ‘. Fine-tuning was performed using the LLaMA-Factory framework [6] on the 700-instance training split. A simpler prompt was used for fine-tuning, focusing on direct entity extraction (see Appendix A.2). Key hyperparameters included a learning rate of 5e-06, 2.0 training epochs, LoRA rank of 16, and LoRA alpha of 32. The final submission used the checkpoint from training step 20, chosen based on preliminary validation.

The JSON output from the LLMs was parsed using a custom function to retrieve the list of entity mentions. We noted occasional JSON parsing errors, a practical challenge when using LLMs for structured data generation, which were handled with fallback logic.

3.2. Entity Linking (EL) - Subtask 2

The EL subtask built upon the entities recognized in Subtask 1. For each extracted Greek entity, we linked it to the most appropriate ICD-10 code from the ‘labelset.txt‘ using semantic similarity with the ‘intfloat/multilingual-e5-large-instruct‘ sentence transformer model [7].

1. **Corpus Preparation**: An ICD-10 code corpus was created by combining the 3-character codes from ‘labelset.txt‘ with their detailed English descriptions from ‘codes.csv‘. Embeddings were pre-computed for each code’s description.
2. **Query Embedding**: For each entity extracted by the LLM, a query embedding was generated using an instructional format. If a contextual relevance description was available (from Config 1 & 2), the prompt was: "Instruct: Given an entity, retrieve the related medical Disease\nQuery: [entity_relevance_english]". If only the Greek entity was available, the prompt was: "Instruct: Given a Greek entity, retrieve the related medical Disease\nQuery: [greek_entity]". This dual-prompting strategy aimed to leverage the richer context when available.
3. **Similarity Matching**: The cosine similarity between the query embedding and all ICD-10 code embeddings was calculated. The ICD-10 code with the highest similarity was assigned.

The three EL configurations (‘config1‘, ‘config2‘, ‘config3‘) directly corresponded to the NER configurations, using their respective entity outputs.

Table 1
Official NER Results for pjmathematician.

System	Precision	Recall	F1-Score
config1	0.2586	0.2484	0.2534
config2	0.2188	0.2892	0.2491
config3 (LoRA)	0.1732	0.3297	0.2271

Table 2
Official EL Results for pjmathematician.

System	Precision	Recall	F1-Score
config1	0.0642	0.0616	0.0629
config2	0.0525	0.0693	0.0597
config3 (LoRA)	0.0112	0.0212	0.0146

3.3. Multi-Label Classification with Explainable AI (MLC-X) - Subtask 3

For the MLC-X subtask, we used a larger, more capable model, ‘Qwen/Qwen2.5-72B-Instruct-AWQ‘, believing its enhanced reasoning abilities would be beneficial for this complex document-level task. A two-stage approach was adopted:

1. **Candidate Generation**: To reduce the search space and guide the LLM, a candidate list of ICD-10 codes was generated for each document. We created embeddings for the original Greek text and its two different English translations (produced by our Config 1 and 2 systems) using ‘multilingual-e5-large-instruct‘. For each of the three texts, we found the top 20 most similar ICD-10 codes from our corpus. The union of these three sets formed the final candidate list for the LLM.
2. **LLM-based Classification and Explanation**: The 72B model was prompted with the Greek text and the filtered list of candidate codes and their descriptions. The prompt (see Appendix A.3) instructed the model to select the relevant codes and extract the exact Greek phrases justifying each selection.

Two final configurations were submitted:

- **Config 4 (MLC with Evidence)**: The direct output of the 72B LLM, including both the predicted ICD-10 codes and their supporting textual evidence.
- **Config 5 (MLC no Evidence)**: From the same LLM output as Config 4, we extracted only the predicted ICD-10 codes and submitted them with mention positions set to -1, as per task guidelines for a classification-only submission.

4. Experiments and Results

The official evaluation was performed on the test set of 500 discharge letters.

4.1. NER and EL Results

Tables 1 and 2 show our official NER and EL results. Config 1 (base Qwen-32B) achieved the highest F1-score for both tasks. The identical Config 2 run produced slightly different results due to the stochastic nature of the generative model, confirming the variability in LLM outputs. The LoRA-tuned model (Config 3) showed higher recall in NER but lower precision, resulting in a lower F1-score. This suggests that while fine-tuning increased sensitivity, it may require more data or a more sophisticated prompt to maintain precision. The EL performance was directly impacted by the upstream NER results, with error propagation leading to lower overall scores.

4.2. MLC-X Results

Table 3 presents our official results for Subtask 3a. For MLC-X, Config 5 (codes only) achieved a higher F1-score than Config 4 (codes with evidence). This is likely because the Subtask 3a evaluation metric penalizes for incorrect mention boundaries. By not providing them, Config 5 avoids this penalty, resulting in a score that purely reflects the code classification accuracy. The stronger performance in this subtask compared to EL suggests that the two-stage approach—using embeddings for candidate filtering and a larger LLM for final classification—is a more effective strategy for document-level coding.

Table 3
Official MLC-X Results for pjmathematician (Subtask 3a).

System	Precision	Recall	F1-Score
config4 (MLC with Evidence)	0.6056	0.2257	0.3288
config5 (MLC no Evidence)	N/A	N/A	0.3655

5. Discussion

Our experiments highlighted several challenges and insights. The primary challenge was the inherent difficulty of clinical NLP in a low-resource language. The performance of EL was critically dependent on the upstream NER task; any errors in entity recognition directly propagated, limiting the potential of the linking module.

The use of different LLM sizes was a conscious design choice. The 32B model was deemed sufficient for the more straightforward (though still challenging) NER task, while the larger 72B model was reserved for the more complex MLC-X reasoning task, which involved selecting from a candidate list and finding evidence. Our results suggest this was a reasonable approach, as the MLC-X F1-scores were considerably higher than the EL scores.

The LoRA fine-tuning experiment (Config 3) yielded interesting results. While it did not outperform the zero-shot base model, it demonstrated a trade-off between precision and recall. With only two training epochs and a small dataset (700 examples), the model may have been under-tuned. Further training or more sophisticated prompt-tuning could potentially improve its performance.

Finally, a practical challenge was the reliability of LLMs in generating perfectly formatted JSON, with several parsing errors encountered during our experiments. This underscores the need for robust post-processing and error-handling when integrating LLMs into structured data extraction pipelines.

6. Conclusion

We presented our systems for the ELCardioCC shared task, leveraging a combination of large language models (Qwen2.5-32B and Qwen2.5-72B), LoRA fine-tuning, and multilingual sentence embeddings. Our approach demonstrated the feasibility of using modern NLP techniques for NER, EL, and MLC-X on Greek clinical text. The results indicate that while zero-shot prompting of capable LLMs provides a strong baseline, the performance of downstream tasks like EL is highly sensitive to NER quality. For document-level classification, a hybrid approach combining semantic retrieval for candidate generation with a powerful LLM for final classification and explanation proved to be the most effective strategy. Future work could explore joint NER and EL models to mitigate error propagation and more extensive fine-tuning to better adapt models to the specific clinical dialect.

Acknowledgments

We thank the organizers of the ELCardioCC shared task for providing the dataset and the evaluation platform. We also acknowledge the developers of the open-source tools and models used in this work, including Hugging Face Transformers, Sentence Transformers, LMDeploy, and LLaMA-Factory.

Declaration on Generative AI

During the preparation of this work, the author(s) used Gemini2.5 Pro in order to: Grammar and spelling check. After using this tool, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] Tsoumakas, G., Giannakoulas, G., Samaras, A., Dimitriadis, D., Patsiou, V., Bekiaridou, A. ELCardioCC: Advancing Clinical Coding in Cardiology: A Challenge on Named Entity Recognition, Entity Linking, Multi-label Classification & Explainable AI. Task Overview. https://elcardiocc.web.auth.gr/

[2] ELCardioCC Shared Task Organizers. Dataset Description. ELCardioCC Website. https://elcardiocc.web.auth.gr/#dataset

[3] Qwen Team. Qwen2.5 Technical Report. https://qwenlm.github.io/blog/qwen1.5/

[4] LMDeploy Contributors. LMDeploy: A High-throughput LLM Inference Engine. GitHub Repository. https://github.com/InternLM/lmdeploy

[5] Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations (ICLR)*, 2022.

[6] LLaMA-Factory Contributors. LLaMA Factory: Unified LLaMA Fine-tuning Framework. GitHub Repository. https://github.com/hiyouga/LLaMA-Factory

[7] Wang, L., Yang, N., Huang, J., Du, X., Wei, F. Text Embeddings by Weakly-Supervised Contrastive Pre-training. *arXiv preprint arXiv:2212.03533*, 2022.

[8] Dimitriadis, D., Patsiou, V., Stoikopoulou, E., Toumpas, A., Kipouros, A., Papadopoulos, D., Bekiaridou, A., Barmpagiannos, K., Vasilopoulou, A., Barmpagiannos, A., Samaras, A., Giannakoulas, G., and Tsoumakas, G. Overview of ElCardioCC Task on Clinical Coding in Cardiology at BioASQ 2025. In *CLEF 2025 Working Notes*, edited by Guglielmo Faggioli, Nicola Ferro, Paolo Rosso, and Damiano Spina, 2025.

A. Prompts Used in Experiments

A.1. Prompt for Base LLM (Config 1 & 2)

This prompt was used for the zero-shot NER and EL tasks with the base ‘Qwen/Qwen2.5-32B-Instruct‘ model.

```
You will be given a Greek medical text. You have to do the following
    :

1. Translate in English
2. Detect Entities (chief complaint, diagnosis, prior medical
   history, drugs and cardiac echo)
3. For every entity give the english translation and also a
   contextual 1-2 sentence description on why it is relevant

Provide the output as JSON inside <JSON> ... </JSON> tags like this:

<JSON>
{
    "english_translation":<english translation of the text here>,
    "extracted_entities":[
        {
            "entity":<extracted entity here (same case exactly how
                it appears in the greek text)>,
            "entity_english_translation":<english translation of the
                entity>,
            "entity_relevance_english":<contextual 1-2 sentence
                description on why it is relevant in English>
        },
        ...
    ]
}
</JSON>
```

A.2. Prompt for LoRA Fine-Tuning and Inference (Config 3)

This simpler system prompt was used for fine-tuning and inference with the LoRA-adapted model, focusing directly on entity extraction.

```
You will be given a Greek medical text. You have to do the following
   :

Detect Entities (chief complaint, diagnosis, prior medical history,
   drugs and cardiac echo). Make sure to include both Greek and
   English entities, ensure everything is covered with high recall.

Provide the output as JSON inside <JSON> ... </JSON> tags like this:

<JSON>
{
    "extracted_entities":[
        {
            "entity":<extracted entity here (same case exactly how
               it appears in the greek text)>,
        },
        ...
    ]
}
</JSON>
```

A.3. Prompt for MLC-X Task (Config 4 & 5)

This prompt was used with the ‘Qwen/Qwen2.5-72B-Instruct-AWQ‘ model for the multi-label classification task. The ‘‘ placeholders were populated with the Greek text and the candidate ICD-10 codes, respectively.

```
# Medical Coding Task: Greek Text to ICD10 Classification

## Your Task
You will analyze Greek medical texts and identify relevant ICD10
   codes from a provided set. For each relevant code, extract the
   specific Greek terms/phrases from the text that justify this
   classification.

## Input
- A Greek medical text (patient record, clinical note, etc.)
- A set of candidate ICD10 codes with their descriptions
## Instructions
1. Carefully read the Greek medical text to understand the clinical
   content
```

```
2. Analyze each provided ICD10 code and determine if it applies to
   the medical text
3. For each relevant ICD10 code, identify and extract the exact
   Greek terms/phrases that support this code assignment
4. Only include ICD10 codes that are clearly supported by the text
5. Extract entities exactly as they appear in the Greek text (
   preserve exact spelling, accents, and form)
6. If no codes are relevant, return an empty ICD10 array

## Output Format
Respond in valid JSON format with this exact structure:
<JSON>
{
    "ICD10": [
        {
            "code": "A25",
            "entities": ["exact greek phrase 1", "exact greek phrase
                2"]
        },
        ...
    ]
}
</JSON>

USER_PROMPT = """
## Greek Medical Text
{}

## ICD10 Codes
{}
"""
```