System Report for CCL25-Eval Task 8: ClinSplitFT: Enhancing ICD Coding in Chinese EMRs with Prompt Engineering and Candidate Set Splitting

Pusheng Chen, Qiangyu Tan, Zhiwen Tang*

Yunnan Key Laboratory of Intelligent Systems and Computing, Yunnan University, Kunming, China

School of Information Science and Engineering, Yunnan University, Kunming, China

chenpusheng@stu.ynu.edu.cn, tanqiangyu@stu.ynu.edu.cn, zhiwen.tang@ynu.edu.cn

Abstract

CCL25-Eval Task 8 focuses on ICD coding from clinical narratives. The challenge of this task lies in the imbalanced and complex label space, with primary diagnoses having a small, focused set of labels and secondary diagnoses involving a much larger, intricate set. To address these challenges, we propose **ClinSplitFT** (**Clin**ical Code **Split F**ine-**T**uning), a novel framework that enhances ICD coding accuracy using large language models (LLMs). The key innovation of ClinSplitFT is its candidate set split strategy, which splits the full candidate set into several manageable subsets and fine-tunes the model separately on each. During inference, predictions from all subsets are aggregated to produce the final output. This split-based fine-tuning approach enables more focused learning and better generalization in multi-label settings, making it an effective solution for clinical code prediction at scale. Experimental results show significant improvements in ICD coding performance. The code for our system is publicly available at https://github.com/277CPS/ICD-Code-prediction.

Keywords: ICD Coding, Candidate Set Splitting, Prompt Engineering, Large Language Model

1 Introduction

The CCL25-Eval Task 8[1] addresses the challenge of automatically assigning International Classification of Diseases (ICDs) codes to Chinese Electronic Medical Records (EMRs). In this task, the model is provided with clinical texts that contain critical information such as the chief complaint, present illness, past medical history, and treatment summary. The goal is to predict both the primary diagnosis code and multiple secondary diagnosis codes, adhering to the ICD-10 standard. The dataset comprises 1,485 de-identified records, spanning 5 primary and 53 secondary diagnostic categories.

This task poses significant challenges due to the imbalanced nature of the label space and the semantic complexity between primary and secondary diagnoses. The primary diagnosis has a smaller, more focused label space, while the secondary diagnoses involve a much larger set of possible labels, each with its own contextual intricacies. These factors complicate both the model's reasoning process and the prediction accuracy.

To tackle these challenges, we propose the ClinSplitFT framework, a novel approach that integrates prompt engineering with candidate set splitting. For primary diagnosis coding, where the label space is limited, we perform direct fine-tuning with structured prompts that guide the model's clinical reasoning, ensuring accurate and focused predictions. In contrast, for secondary diagnosis coding, where the label space is large and varied, we introduce a candidate set split strategy. This approach involves dividing the candidate ICD codes into smaller, manageable subsets. The model generates independent predictions for each subset, and the final output is aggregated from all predictions. This method effectively reduces inference complexity and enables the model to focus on specific subspaces, enhancing both precision and efficiency.

[1] https://tianchi.aliyun.com/competition/entrance/532302

Experimental results demonstrate that ClinSplitFT achieves significant improvements in both single-label and multi-label ICD coding tasks, providing a scalable and effective solution for ICD prediction in real-world clinical applications.

2 Background

2.1 Task Definition

Given a structured clinical record $\mathbf{x}$ in JSON format, comprising fields such as chief complaint, past medical history, physical examination, and other relevant sections. We first linearize the input into a single natural language passage. The goal is to perform *diagnosis coding*, i.e., automatically assign diagnostic codes that reflect the patient's primary and secondary conditions.

Formally, the model learns a mapping function:

$$f : \mathbf{x} \rightarrow \mathcal{O}$$

where $\mathcal{O}$ denotes the output space of formatted diagnostic code strings:

$$[\text{primary} \mid \text{secondary}_1; \text{secondary}_2; \ldots; \text{secondary}_n]$$

- primary $\in \mathcal{Y}_1$: the uniquely assigned primary diagnosis code, selected from the candidate set $\mathcal{Y}_1$.
- secondary$_i$ $\in \mathcal{Y}_2$: the i-th secondary diagnosis code ($n \geq 1$), selected from the candidate set $\mathcal{Y}_2$.

2.2 Related Work

Predicting ICD codes from Chinese EMRs is challenging due to the unstructured nature of clinical texts, linguistic nuances, and regional terminology variations (Zhang et al., 2019). Early rule-based and traditional machine learning approaches, like support vector machines, struggled with these complexities, particularly in capturing long-range dependencies in lengthy narratives (Wang et al., 2020). While Transformer-based models like BERT improved performance by modeling contextual relationships (Liu et al., 2022), challenges persist in multi-label secondary diagnosis coding and handling rare diseases with limited data (Chen et al., 2024). Recent advances using large language models (LLMs), prompt engineering, and parameter-efficient fine-tuning (e.g., LoRA) (Hu et al., 2021) address these issues, but Chinese EMRs still pose unique difficulties for accurate and efficient ICD coding.

3 System

The ClinSplitFT framework consists of three key modules: Prompt Engineering, Primary Diagnosis Coding, and Secondary Diagnosis Coding. The overall framework is illustrated in Figure 1.

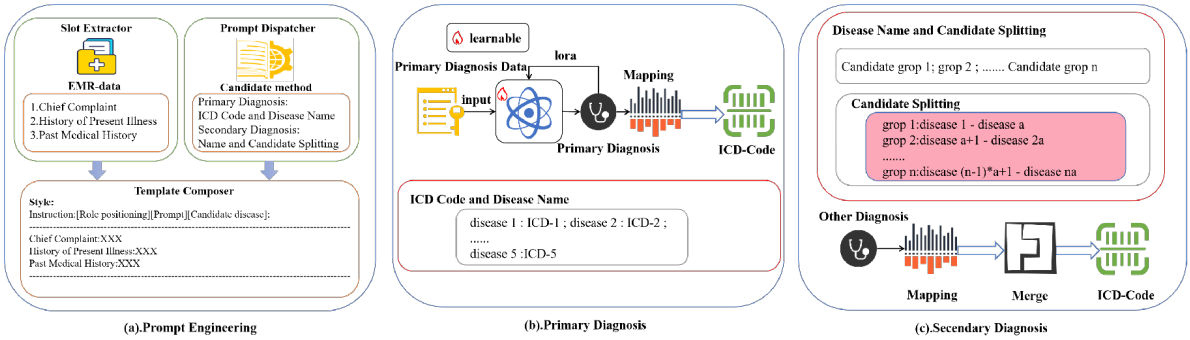

Figure 1: The framework of ClinSplitFT

3.1 Prompt Engineering

We designed a Prompt Engineering module that transforms unstructured EMRs into semantically rich prompts tailored for different coding layers. This module consists of three key components, Slot Extractor, Template Composer, and Prompt Dispatcher, which work in tandem to support hierarchical ICD coding.

The Slot Extractor is responsible for retrieving critical clinical information from patient cases. It targets structured fields such as chief complaint, history of present illness, and past medical history, aligning each with predefined semantic slots in the prompt template. In practice, this extraction process follows a rule, based matching strategy, where each slot is automatically populated based on the corresponding field in the input JSON record.

The Prompt Dispatcher determines the coding stage, primary diagnosis category or secondary fine grained code, and selects the appropriate prompting strategy. For the first level coding, the composed prompt is used to directly fine-tune the LLM. In contrast, for the second level coding, a candidate splitting strategy is activated. This strategy splits the candidate ICD codes into some chunks, and the generated prompt is routed through multiple parallel prompts, each corresponding to a specific chunk. This hierarchical engineering mechanism prevents the model from being overwhelmed by an overly large label space and ensures that its attention is focused on a more manageable and relevant prediction scope.

The Template Composer injects the extracted slot content into predefined prompt templates. These templates are carefully designed to be compatible with the input format expectations of LLMs. By adopting a standardized LLM, friendly structure, the templates enhance the model's ability to interpret the clinical narrative, helping it focus on relevant contextual cues while maintaining generalization across diverse cases.

Prompt for Primary Diagnosis Coding
Instruction: You are a medical expert. Based on the patient's clinical information, identify the most appropriate primary diagnosis code from the candidate list. Consider the patient's chief complaint, present illness, past medical history, admission diagnosis, and so on. Only one primary code should be selected: Candidate Diagnoses: Hypertension grade 3 (very high risk): I10.x00x032, Unstable angina: I20.000, Microvascular angina: I20.800x007, Acute non-ST elevation myocardial infarction: I21.401, Chronic heart failure exacerbation: I50.900x018. **Input:** Chief complaint: Chest tightness. Present illness: Activity-induced symptoms. Past medical history: Hypertension, hyperlipidemia, thyroid surgery. Admission diagnosis: Coronary artery disease, hypertension, hyperlipidemia. Treatment process: Coronary angiography, artery plaque. **Output:** Microvascular angina: I20.800x007

Table 1: Example Prompt for Primary Diagnosis Coding

3.2 Primary Diagnosis

The primary diagnosis coding focuses on identifying the primary diagnostic category, a task that requires robust generalization while preserving domain-specific sensitivity. While LLMs demonstrate impressive zero-shot and few-shot capabilities across a broad range of natural language tasks, their performance often deteriorates when applied to specialized medical domains due to vocabulary mismatches, limited exposure to clinical expressions, and the nuanced reasoning required for diagnostic inference.

To address this, we cast primary diagnosis coding as a domain adaptation problem. Specifically, we leverage the previously composed, semantically structured prompts (see Section 3.1) to guide the model through clinically meaningful reasoning patterns. These prompts not only introduce structured clinical content in a format that LLMs can readily consume, but also mitigate the risk of misunderstanding due to non-standardized EMR narratives.

Examples of prompts utilized during the Primary Diagnosis Coding phase are presented in Table 1.

Prompt for Secondary Diagnosis Coding
Instruction: You are a medical expert. Based on the patient's clinical information, identify all applicable secondary diagnosis codes from the candidate list. Consider the patient's patient's chief complaint, present illness, past medical history, admission diagnosis, and so on. Multiple codes may be selected. Candidate Diagnoses (Split 1–13): Thyroid nodule: E04.101, Thyroid cyst: E04.102, Type 2 diabetes mellitus: E11.900, Diabetes mellitus: E14.900x001, Hyperhomocysteinemia: E72.101, Hyperlipidemia: E78.500, Hypokalemia: E87.600, Hypertension grade 1 (high risk): I10.x00x023, ... **Input:** Chief complaint: Chest tightness. Present illness: Activity-induced symptoms. Past medical history: Hypertension, hyperlipidemia, thyroid surgery. Admission diagnosis: Coronary artery disease, hypertension, hyperlipidemia. Treatment process: Coronary angiography, artery plaque. **Output:** Diabetes mellitus: E14.900x001, Hypokalemia: E87.600

Table 2: Multi-Split Secondary Diagnosis Coding Example

3.3 Secondary Diagnosis

The secondary diagnosis stage demands more fine-grained reasoning over a significantly larger label space. Each clinical case may involve multiple comorbidities, complications, or related conditions, making this task substantially more complex in both semantics and scale. To address this, we propose a Candidate Set split strategy that constrains the inference scope of LLMs without sacrificing diagnostic coverage.

Specifically, instead of exposing the LLM to the entire set of secondary ICD codes, we split the candidate space into subsets of similar size: ICD codes 1-13 are divided into subset 1, ICD codes 14-26 are categorized into subset 2, ICD codes 27-39 form subset 3, and ICD codes 40-53 constitute subset 4. For each subset, a dedicated prompt is constructed and routed into the model, enabling the LLM to perform inference within a limited and focused candidate space.

By narrowing the candidate scope, the model can concentrate more precisely on relevant diagnostic possibilities, reducing noise and potential confusion between similar codes. This split strategy allows the model to process long candidate lists across multiple rounds of prompting, effectively avoiding context window overflows.

Moreover, if the ground-truth label is not included in a given candidate subset, the model will produce no prediction for that subset. This data will be excluded during the final aggregation phase. Only subsets that yield predictions are aggregated to form the final output.

We further fine-tune the model using these decomposed prompts. In contrast to the Primary Diagnosis Coding, this fine-tuning process emphasizes multi-label generation, as a single case may be associated with multiple secondary diagnosis codes.

Examples of prompts utilized during the Secondary Diagnosis Coding phase are presented in Table 2, which only considers the first 13 candidate codes.

4 Experimental

4.1 Data and Evaluation Merics

The system was conducted on Chinese EMR datasets: 800 samples for training, and 200 for testing. The task is structured in two hierarchical Coding stages. In the first stage, the model directly predicts the primary diagnosis code. For the second stage, which involves predicting secondary diagnosis codes from a larger candidate set, we employ a splited fine-tuning strategy. Specifically, the training set for the second stage is expanded to 3,200 samples by dividing the full candidate label space into four subsets and fine-tuning separately on each.

It is important to note that the validation set does not contain ground-truth labels. As a result, during training, only the training set is utilized for model fine-tuning.

We adopt an accuracy-based metric that jointly considers the correctness of the primary diagnosis code and the quality of the secondary diagnosis code predictions. The overall evaluation metric, denoted as Acc, is defined as:

$$Acc = \frac{1}{N}\sum_{i=1}^{N}\{0.5 \cdot I(\hat{y}_{\text{main}} = y_{\text{main}}) + 0.5 \cdot F1(\hat{y}_{\text{other}}, y_{\text{other}})\}_i \tag{1}$$

where:

- N is the total number of test instances.
- y_{main} and $\hat{y}_{\text{main}}$ denote the ground truth and predicted primary diagnosis code, respectively.
- y_{other} and $\hat{y}_{\text{other}}$ represent the sets of ground truth and predicted secondary diagnosis codes, respectively.
- $I(\cdot)$ is an indicator function that returns 1 if the argument is true, and 0 otherwise.
- $F1(\cdot)$ denotes the F1-score calculated between the predicted and true secondary diagnosis code sets.

The F1-score used for evaluating secondary diagnosis codes is computed as follows:

$$F1(\hat{y}, y) = 2 \cdot \frac{Precision(\hat{y}, y) \cdot Recall(\hat{y}, y)}{Precision(\hat{y}, y) + Recall(\hat{y}, y)} \tag{2}$$

Here, $Precision(\hat{y}, y) = \frac{\text{NUM}(\hat{y} \cap y)}{\text{NUM}(\hat{y})}$ and $Recall(\hat{y}, y) = \frac{\text{NUM}(\hat{y} \cap y)}{\text{NUM}(y)}$, where $\hat{y}$ denotes the set of predicted diagnosis code, and y denotes the ground-truth diagnosis code.

4.2 Implementation details

We implemented our method on Mistral-7B (Jiang et al., 2023), Qwen2.5-7B (Yang et al., 2025b) and Qwen3-4B (Yang et al., 2025a). We fine-tuned the LLM using the PEFT (Parameter-Efficient Fine-Tuning) method, more precisely LoRA (Hu et al., 2021). Experiments were conducted on RTX 4090 GPU with LLaMA-Factory (Zheng et al., 2024) api, batch size 1, learning rate 3e-5, and 5 epochs. Training/inference took about 3.5/2.5 hours.

4.3 Results

Table 3 and Table 4 show the results of our experiments. We observe that our proposed method, ClinSplitFT, consistently improves diagnostic performance across both primary and secondary tasks, regardless of the underlying language model. Notably, the integration of the candidate set splitting strategy

demonstrates particular effectiveness in the secondary diagnosis scenario, where both the input and output spaces are inherently large and complex due to the multi-label nature of the task and the rich clinical information involved. Additionally, ablation studies confirm the critical role of fine-tuning. The results highlight the effectiveness of our candidate set split strategy in managing the complexity of secondary diagnosis prediction. By spliting the full ICD code space into subsets, the model is relieved from reasoning over an overwhelming label space, enabling more focused inference. This targeted decomposition not only mitigates noise and code ambiguity but also allows the model to operate efficiently within the context length constraints.

The effectiveness of candidate set decomposition primarily stems from reducing the search space exposed to the model during each inference round. Large language models often struggle when tasked with selecting from an extensive label set due to limitations in context length and output precision. By splitting the secondary diagnosis space into manageable subsets, our method ensures that the model operates within a narrower, computationally feasible scope. This not only mitigates the risk of context overflow but also reduces output ambiguity by allowing the model to focus on a smaller set of clinically relevant options at a time.

Primary Diagnosis			
Method	Qwen3	Qwen2.5	Mistral
ClinSplitFT	**89.97**	74.34	81.45
Secondary Diagnosis			
ClinSplitFT	**66.33**	55.98	52.87

Table 3: Performance Comparison on Primary and Secondary Diagnoses

Primary Diagnosis			
Method	Qwen3	Qwen2.5	Mistral
ClinSplitFT	**89.97**	74.34	81.45
ClinSplitFT(w/o FT)	28.23	25.38	27.89
Secondary Diagnosis			
ClinSplitFT	**66.33**	62.53	60.87
ClinSplitFT(w/o ClinSplit)	60.73	60.45	40.78
ClinSplitFT(w/o FT)	9.35	8.09	7.65

Table 4: Ablition Study

It is important to note that the italicized values in the tables are derived from evaluation on the training set. For fine-tuning-based methods, we used 80% of the training data for training and the remaining 20% for testing. For methods that do not involve fine-tuning, the entire training set was used for evaluation. This setup was necessary due to the limited number of allowed submissions and the unavailability of ground-truth labels for the official test set.

5 Conclusion

This paper presents **ClinSplitFT**, a framework designed to address the challenges of ICD coding in Chinese Electronic Medical Records (EMRs) through candidate set split and task-specific fine-tuning. ClinSplitFT adopts distinct fine-tuning strategies for primary and secondary diagnosis codes: it applies direct fine-tuning for primary diagnosis prediction, while for secondary diagnosis prediction, it introduces a split-based fine-tuning approach. By partitioning the candidate code set into specialized subsets,

the split-based method enables more focused model adaptation and mitigates interference during multi-label prediction. This approach underscores the importance of aligning large language model fine-tuning strategies with the structural characteristics of medical coding tasks, providing a scalable and practical solution for handling complex diagnostic code systems in clinical settings.

Acknowledgements

This work was supported by the Open Project Program of Yunnan Key Laboratory of Intelligent Systems and Computing (Grant No. ISC24Y03) and the Yunnan Fundamental Research Project (Grant No. 202501AT070231).

References

Yun Zhuang, Jing Zhang, Xin Li, Chen Liu, Yiwen Yu, Wei Dong, and Kevin He. Autonomous International Classification of Diseases Coding Using Pretrained Language Models and Advanced Prompt Learning Techniques: Evaluation of an Automated Analysis System Using Medical Text. *JMIR Medical Informatics*, 13:e63020, 2025.

Jian Wang, Xiaobo Zhang, Fei Wang, and Kexin Zhang. Intelligent Diagnosis with Chinese Electronic Medical Records Based on Convolutional Neural Networks. *BMC Bioinformatics*, 20:62, 2020.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models. arXiv preprint arXiv:2106.09685, 2021.

Xiaobo Zhang, Jian Wang, Fei Wang, and Kexin Zhang. Intelligent Diagnosis with Chinese Electronic Medical Records Based on Convolutional Neural Networks. *BMC Bioinformatics*, 20:62, 2019.

Yang Liu, Yining Chen, and Hao Zhou. Transformer-Based Models for Automated ICD Coding: A Comparative Study on Chinese and English Clinical Texts. *Artificial Intelligence in Medicine*, 128:102312, 2022.

Chen Li, Hongyu Liu, and Wei Lu. Deep Learning for Multilingual Clinical Entity Recognition in Electronic Health Records. *Journal of Biomedical Informatics*, 135:104215, 2023.

Tao Chen, Ming Zhang, and Xiaodong Wu. Prompt-Based Fine-Tuning for Low-Resource Medical Text Classification. *IEEE Journal of Biomedical and Health Informatics*, 28(3):1122–1133, 2024.

Qinwei Xu, Zhian Bai, Yuchen Xu, Xingkun Xu, and Chenyi Zhou. Towards Normalized Clinical Information Extraction in Chinese Radiology Report with Large Language Models. *ScienceDirect*, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, et al. Qwen3 Technical Report. arXiv preprint arXiv:2505.09388, 2025.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, et al. Qwen2.5 Technical Report. arXiv preprint arXiv:2412.15115, 2025.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, et al. Mistral 7B. arXiv preprint arXiv:2310.06825, 2023.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. LlamaFactory: Unified Efficient Fine-Tuning of 100+ Language Models. arXiv preprint arXiv:2403.13372, 2024.