Autonomous International Classification of Diseases Coding Using Pre-Trained Language Models and Advanced Prompt Learning Techniques

Yan Zhuang, Junyan Zhang, Xiuxing Li, Chao Liu, Yue Yu, Wei Dong, Kunlun He

Submitted to: JMIR Medical Informatics
on: June 11, 2024

Disclaimer: © The authors. All rights reserved. This is a privileged document currently under peer-review/community review. Authors have provided JMIR Publications with an exclusive license to publish this preprint on it's website for review purposes only. While the final peer-reviewed paper may be licensed under a CC BY license on publication, at this stage authors and publisher expressively prohibit redistribution of this draft paper other than for review purposes.

Table of Contents

Autonomous International Classification of Diseases Coding Using Pre-Trained Language Models and Advanced Prompt Learning Techniques

Yan Zhuang[1*] PhD; Junyan Zhang[1*] M.E; Xiuxing Li[2] PhD; Chao Liu[3] PhD; Yue Yu[3] BS; Wei Dong[4] PhD; Kunlun He[1] PhD

[1]Medical Big Data Research Center Chinese PLA General Hospital Beijing CN
[2]School of Computer Science & Technology Beijing Institute of Technology Beijing CN
[3]Digital Health China Technologies Co., Ltd. Beijing CN
[4]Senior Department of Cardiology the Sixth Medical Center of PLA General Hospital Beijing CN
[*]these authors contributed equally

Corresponding Author:
Kunlun He PhD
Medical Big Data Research Center
Chinese PLA General Hospital
28 Fuxing Road
Beijing
CN

Abstract

Background: Due to the limitations posed by small datasets, diverse writing styles, unstructured clinical records, and the necessity of semi-manual preprocessing, machine learning techniques for real-time ICD coding continue to face significant challenges.

Objective: In this study, we developed a fully automatic pipeline from long free text to standard ICD codes, which integrated medical pre-trained and keyword filtration BERT, fine-tuning, and task-specific prompt learning with mixed templates and soft verbalizers.

Methods: We integrated four components into our framework: a medical pre-trained BERT, a keyword filtration BERT, a fine-tuning phase, and task-specific prompt learning which utilized mixed templates and soft verbalizers. This framework was validated on a multi-center medical dataset for the automated ICD coding of 13 common cardiovascular diseases. Its performance was compared against RoBERTa, XLNet, and different BERT-based fine-tuning pipelines. Additionally, we evaluated the performance of our framework under different prompt learning and fine-tuning settings. Further, few-shot learning was conducted to assess the feasibility and efficacy of our framework in scenarios involving small to mid-sized datasets.

Results: Compared to traditional pre-training and fine-tuning pipelines, our approach achieved a significantly higher micro-F1 score of 0.838 and a macro-AUC of 0.958. Among different prompt learning setups, the mixed template and soft verbalizer combination yielded the best performance. Few-shot experiments indicated that performance stabilized and peaked at 500 shots.

Conclusions: These findings underscore the effectiveness and superior performance of prompt learning and fine-tuning for subtasks within pre-trained language models in medical practice. Our real-time ICD coding pipeline effectively extracts detailed medical free-text into standardized labels, with potential applications in clinical decision-making.

(JMIR Preprints 11/06/2024:63020)
DOI: https://doi.org/10.2196/preprints.63020

2) If accepted for publication in a JMIR journal, would you like the PDF to be visible to the public?

✓ **Yes, please make my accepted manuscript PDF available to anyone at any time (Recommended).**

Yes, but please make my accepted manuscript PDF available only to logged-in users; I understand that the title and abstract will remain v

Yes, but only make the title and abstract visible (see Important note, above). I understand that if I later pay to participate in <a href="http

Original Manuscript

Autonomous International Classification of Diseases Coding Using Pre-Trained Language Models and Advanced Prompt Learning Techniques

Authors and Affiliations

Yan Zhuang[1,*], Junyan Zhang[1,*], Xiuxing Li[2], Chao Liu[3], Yue Yu[3], Wei Dong[4], Kunlun He[1,#]

1. Medical Big Data Research Center, Chinese PLA General Hospital, Beijing, China
2. School of Computer Science & Technology, Beijing Institute of Technology, China
3. Digital Health China Technologies Co., Ltd.
4. Senior Department of Cardiology, the Sixth Medical Center of PLA General Hospital, Beijing, China

*Yan Zhuang and Junyan Zhang contributed equally as the co-firtst author.

#Corresponding auther: Kunlun He
Email: kunlunhe@plagh.org
Affiliations:

1. Medical Big Data Research Center, Chinese PLA General Hospital, Beijing, China

Address: No.28, Fuxing Road, Haidian District, Beijing, China, 100039

Abstract

Background:

Due to the limitations posed by small datasets, diverse writing styles, unstructured clinical records, and the necessity of semi-manual preprocessing, machine learning techniques for real-time ICD coding continue to face significant challenges.

Objective:

In this study, we developed a fully automatic pipeline from long free text to standard ICD codes, which integrated medical pre-trained and keyword filtration BERT, fine-tuning, and task-specific prompt learning with mixed templates and soft verbalizers.

Methods:

We integrated four components into our framework: a medical pre-trained BERT, a keyword filtration BERT, a fine-tuning phase, and task-specific prompt learning which utilized mixed templates and soft verbalizers. This framework was validated on a multi-center medical dataset for the automated ICD coding of 13 common cardiovascular diseases. Its performance was compared against RoBERTa, XLNet, and different BERT-based fine-tuning pipelines. Additionally, we evaluated the performance of our framework under different prompt learning and fine-tuning settings. Further, few-shot learning was conducted to assess the feasibility and efficacy of our framework in scenarios involving small to mid-sized datasets.

Results:

Compared to traditional pre-training and fine-tuning pipelines, our approach achieved a significantly higher micro-F1 score of 0.838 and a macro-AUC of 0.958. Among different prompt learning setups, the mixed template and soft verbalizer combination yielded the best performance. Few-shot experiments indicated that performance stabilized and peaked at 500 shots.

Conclusions:

These findings underscore the effectiveness and superior performance of prompt learning and fine-tuning for subtasks within pre-trained language models in medical practice. Our real-time ICD coding pipeline effectively extracts detailed medical free-text into standardized labels, with potential applications in clinical decision-making.

Keywords: BERT; pre-trained language models; prompt learning; ICD; cardiovascular disease; few-shot learning; multi-center medical data

Introduction

Background

The International Classification of Diseases, 10th Revision (ICD-10), is a universally recognized diagnostic categorization system, widely used in medical insurance reimbursements, health reporting, mortality assessments, and related fields [1]. The ICD-10 automatic coding mechanism facilitates swift and accurate classification and statistical analysis of medical data, providing a scientific basis for effective hospital administration and decision-making. In addition, the ICD-10 automatic coding system accelerates disease diagnosis and treatment planning for medical practitioners, thereby enhancing medical efficacy and quality. Therefore, achieving precise ICD coding remains a crucial concern in clinical practice.

In hospital settings, the assignment of ICD codes to unstructured clinical narratives in medical records is a manual task performed by skilled medical coders based on the attending physician's clinical diagnosis. Despite its essential role, this process is plagued by inefficiencies such as time consumption, susceptibility to errors, and high costs. Furthermore, manual coding cannot always guarantee the accuracy of ICD coding due to the complexity of code assignment, which requires a comprehensive consideration of the patient's overall health condition, including medical history, co-existing medical conditions, complications, surgical interventions, and specialized diagnostic procedures [2, 3].

Machine Learning Techniques

The demand to enhance efficiency and minimize errors has spurred the development of numerous machine learning techniques, which help automate the process of medical ICD coding. These techniques can be categorized into four main types: rule-based systems [4, 5], traditional supervised algorithms [6, 7], gate unit-based deep learning [7-9], and pre-trained language models (PLMs) [9-16].

Firstly, rule-based systems for automatic ICD coding involve the formulation of explicit rules and knowledge bases to map medical records to appropriate ICD codes [4, 5]. Although these approaches have been in use for decades and have laid the groundwork for more sophisticated techniques, they suffer from limited adaptability and scalability.

Secondly, traditional supervised algorithms, such as gradient boosted trees, have been employed for ICD coding due to their ability to handle large-scale, high-dimensional datasets efficiently after semi-structured preprocessing, which involves organizing and refining semi-structured data into a usable format [6, 7]. For instance, Diao et al. constructed a LightGBM-based pipeline for auto-coding 168 primary diagnosis ICD-10 codes in discharge records and procedure texts, achieving an accuracy of 95.2% [6]. Another study combined Long Short-Term Memory (LSTM) with attention mechanisms to predict mortality in ICU patients based on their electronic health records (EHRs), demonstrating a much higher AUC score than traditional statistical models and LSTM alone [7].

Thirdly, PLMs represent a neural network model with a fixed architecture trained on a large corpus, which can be fine-tuned for specific downstream tasks such as question answering and entity recognition [10-13]. Bidirectional Encoder Representations from Transformers (BERT) is a prominent PLM designed to learn deep bidirectional representations from large-scale unlabeled text data, capturing semantic relationships in clinical records and easily adapting to various natural language processing (NLP) tasks through task-specific layers [13]. Coutinho and Martins proposed a BERT model with a fine-tuning method for automatic ICD-10 coding of death certificates based on free-text descriptions and associated documents [14]. Moreover, Yan et al. introduced RadBERT, an ensemble model of BERT-base, Clinical-BERT, robustly optimized BERT pretraining approach (RoBERTa), and BioMed-RoBERTa adapted for radiology. Liu, et al. evaluated RadBERT on three NLP tasks: abnormal sentence classification, report coding, and report summarization, demonstrating significantly better performance than existing transformer language models [15, 17].

Finally, XLNet is another method of PLMs that can capture both forward and backward contexts of text [16]. It combines the advantages of autoregressive (AR) models and autoencoding (AE) models while avoiding their limitations. XLNet uses a permutation-based objective function that maximizes the expected likelihood of a text over all possible orderings of its words. It also incorporates the Transformer-XL architecture, allowing for long-term dependency modeling and memory efficiency. XLNet has been reported to outperform BERT and other baselines on several natural language understanding tasks.

Prompt Engineering Techniques

On the other hand, prompt engineering is a technique that involves the careful construction of prompts or inputs for AI models to enhance their performance on specific tasks. This technique encompasses the selection of appropriate words, phrases, symbols, and formats to guide a large language model in generating high-quality and relevant texts. Numerous studies have employed prompts for model tuning to bridge the gap between pre-training objectives and downstream tasks, showing that both discrete and continuous prompts can induce better performance in few-shot and zero-shot tasks [18, 19]. Additionally, this technique within PLMs has been shown to outperform fine-tuning in various clinical decision tasks [20]. It offers the advantage of requiring less data and computational resources, making it particularly suitable for clinical settings.

There are two primary categories of prompting methods: hard prompts and soft prompts [20-23]. Hard prompts, where the prompt is an actual text string, involve methods that automatically search for templates described in a discrete space such as mining-based, paraphrasing-based, and gradient-based approaches [24-26]. The advantages of hard prompts include their interpretability, portability, flexibility, and simplicity. However, creating effective prompts for specific tasks requires substantial effort and creativity.

Soft prompts, on the other hand, are learnable tensors concatenated with the input embeddings, which can be optimized for a given dataset. The key advantage of soft prompts is their ability to achieve better performance than hard prompts by adapting to the model and the data. However, they are not human-readable and lack portability across different models.

Prefix tuning and P-tuning are two methods of prompt engineering that can improve performance beyond traditional fine-tuning [20-22]. Prefix tuning is a lightweight method that keeps the PLM parameters unchanged while optimizing a sequence of task-specific vectors called the prefix [21]. This prefix is added to the input and interacts with the model's hidden states at each layer. Its success depends on how well the prefix is initialized, especially when there is limited data. P-tuning is another prompt tuning strategy that performs as well as fine-tuning across various tasks [22]. It reduces the number of PLM parameters through self-adaptive pruning and tunes a few continuous prompts at the beginning of each transformer layer.

The verbalizer is the final layer that defines the answer space and maps it to the target output. Typically, verbalizers are manually crafted, which can limit their coverage due to personal vocabulary biases [19, 27]. Thus, some studies have proposed automatic verbalizer searching methods to identify better verbalizers, also known as soft verbalizers [18, 28-30].

Autonomous ICD Coding in CVD

Nowadays, cardiovascular disease (CVD) is a leading cause of death worldwide, presenting a high risk of mortality among patients [7]. Automatically labeling patients with CVD is crucial for clinical decision-making and resource allocation. However, existing prediction models face limitations such as low accuracy, lack of generalizability, and inability to capture multi-center data. To address these challenges, we propose a prompt learning real-time framework based on PLMs that can automatically label long free text with ICD-10 codes for CVDs without semi-automatic preprocessing.

Our framework comprises four components: a medically-oriented pre-trained BERT, a keyword filtration BERT, a fine-tuning phase, and task-specific prompt learning facilitated by mixed templates and soft verbalizers. To validate the efficacy of our framework, we conducted comprehensive evaluations on a Chinese multi-center cardiovascular dataset encompassing data from 13,000 CVD patients. This deliberate choice of dataset ensures the robustness and broad applicability of our framework. We compared our framework with RoBERTa, XLNet, and various BERT-based fine-tuning pipelines to highlight its performance. Additionally, we performed few-shot experiments to show its resilience. This pursuit promises to yield valuable insights into the enhancement of medical knowledge extraction and its effective application, thereby warranting continued research and development in this promising domain. In future work, we plan to implement this fully automated ICD coding pipeline across various clinical applications, including clinical decision support systems, cohort studies, and disease early warning and diagnosis systems.

Method

Overview

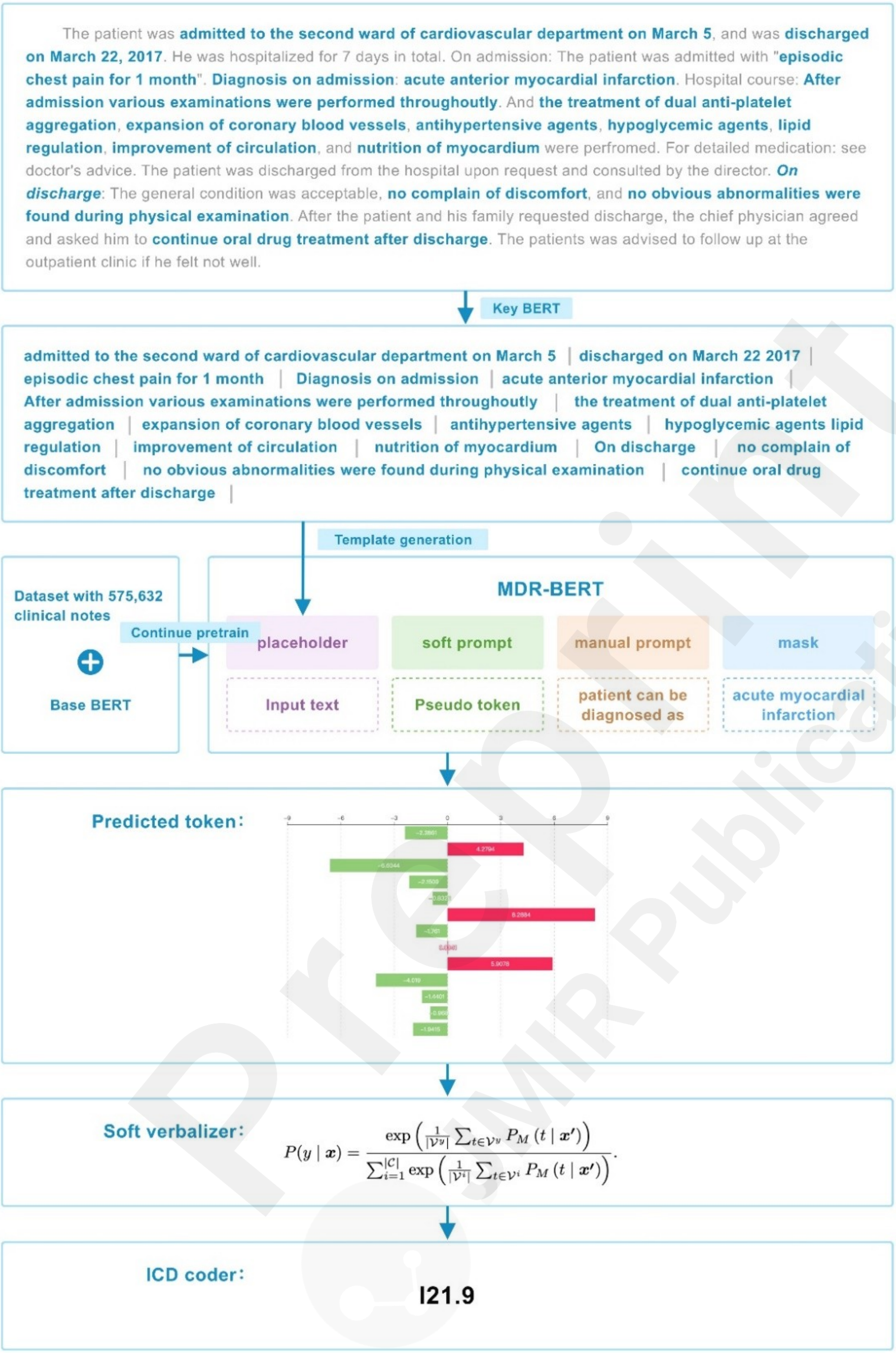
The overall framework of the model is shown in Figure 1. We used a corpus dataset of 575,632 clinical notes to continue training the original BERT model as the PLM for our work, named MDR-BERT. For the classification task, we first applied key-BERT to filter the discharge summaries, which extracts key words and splits the long free text into short sentences.

We then composed the input template for fine-tuning and prompt learning using three components: soft prompt, manual prompt, and mask part. The manual prompt was a handcrafted text prompt with discrete tokens. The soft prompt was a learnable pseudo-token with a few continuous parameters. The mask part was the ICD coding label. Finally, we used a trainable soft verbalizer to compute and apply the softmax function to the probabilities of ICD classes, producing the output.

Dataset Characteristics

The cardiovascular dataset used in this study was obtained from the Cardiovascular Department of Chinese PLA General Hospital's Medical Big Data Research Center, Beijing, China, from 2017 to 2021. To ensure privacy, patient names and addresses were desensitized. The data platform consists of EHRs aggregated from eight affiliated medical centers. A total of 584,969 clinical notes with structured ICD labels were extracted from admission records and discharge summaries in the Cardiovascular Department. The detailed distribution and basic statistical information of the dataset are shown in Figure 2.

Based on the long-tailed distribution and clinician selection, 13 diseases were selected for classification. These diseases include atrial fibrillation (AF), acute myocardial infarction (AMI), infective endocarditis (IE), acute left heart failure (ALHF), acute coronary syndrome (ACS), acute aortic dissection (AAD), hypertensive emergency (HE), acute pulmonary embolism (APE), acute myocarditis (AM), ventricular tachycardia (VT), cardiogenic shock (CS), acute heart failure (AHF), and third-degree atrioventricular block (TAB). Their corresponding ICD-10 codes and abbreviations are listed in Table 1.

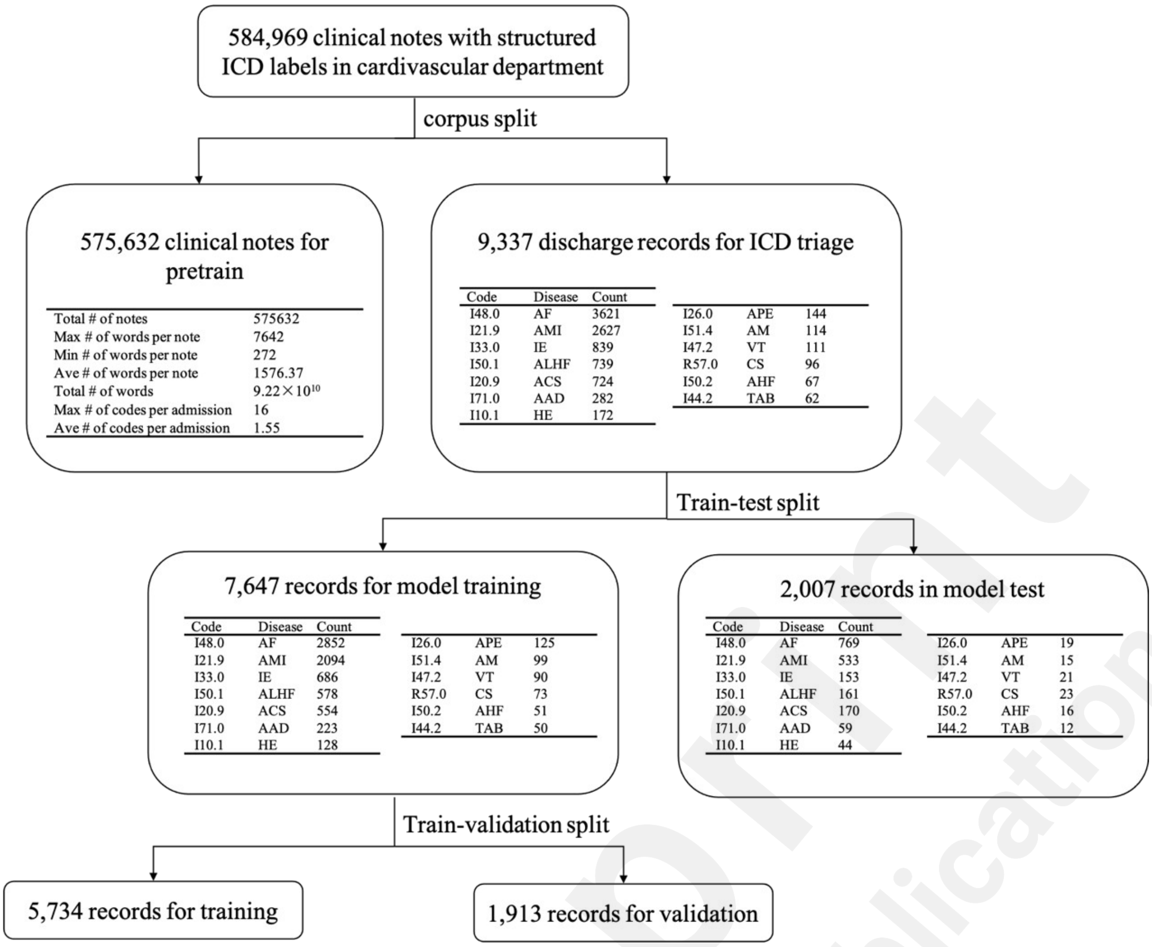


The patient was admitted to the second ward of cardiovascular department on March 5, and was discharged on March 22, 2017. He was hospitalized for 7 days in total. On admission: The patient was admitted with "episodic chest pain for 1 month". Diagnosis on admission: acute anterior myocardial infarction. Hospital course: After admission various examinations were performed throughoutly. And the treatment of dual anti-platelet aggregation, expansion of coronary blood vessels, antihypertensive agents, hypoglycemic agents, lipid regulation, improvement of circulation, and nutrition of myocardium were perfromed. For detailed medication: see doctor's advice. The patient was discharged from the hospital upon request and consulted by the director. *On discharge*: The general condition was acceptable, no complain of discomfort, and no obvious abnormalities were found during physical examination. After the patient and his family requested discharge, the chief physician agreed and asked him to continue oral drug treatment after discharge. The patients was advised to follow up at the outpatient clinic if he felt not well.

admitted to the second ward of cardiovascular department on March 5 | discharged on March 22 2017 | episodic chest pain for 1 month | Diagnosis on admission | acute anterior myocardial infarction | After admission various examinations were performed throughoutly | the treatment of dual anti-platelet aggregation | expansion of coronary blood vessels | antihypertensive agents | hypoglycemic agents lipid regulation | improvement of circulation | nutrition of myocardium | On discharge | no complain of discomfort | no obvious abnormalities were found during physical examination | continue oral drug treatment after discharge |

$$P(y \mid \boldsymbol{x})=\frac{\exp \left(\frac{1}{\left|\mathcal{V}^{y}\right|} \sum_{t \in \mathcal{V}^{y}} P_{M}\left(t \mid \boldsymbol{x}^{\prime}\right)\right)}{\sum_{i=1}^{|\mathcal{C}|} \exp \left(\frac{1}{\left|\mathcal{V}^{i}\right|} \sum_{t \in \mathcal{V}^{i}} P_{M}\left(t \mid \boldsymbol{x}^{\prime}\right)\right)} .$$

Figure 1. Overall framework of MDR-BERT, key-BERT, prompt learning pipeline.


584,969 clinical notes with structured ICD labels in cardivascular department
corpus split
575,632 clinical notes for pretrain
9,337 discharge records for ICD triage
Train-test split
7,647 records for model training
2,007 records in model test
Train-validation split
5,734 records for training
1,913 records for validation

Total # of notes	575632
Max # of words per note	7642
Min # of words per note	272
Ave # of words per note	1576.37
Total # of words	9.22×10^{10}
Max # of codes per admission	16
Ave # of codes per admission	1.55

Code	Disease	Count			
I48.0	AF	3621	I26.0	APE	144
I21.9	AMI	2627	I51.4	AM	114
I33.0	IE	839	I47.2	VT	111
I50.1	ALHF	739	R57.0	CS	96
I20.9	ACS	724	I50.2	AHF	67
I71.0	AAD	282	I44.2	TAB	62
I10.1	HE	172			

Code	Disease	Count			
I48.0	AF	2852	I26.0	APE	125
I21.9	AMI	2094	I51.4	AM	99
I33.0	IE	686	I47.2	VT	90
I50.1	ALHF	578	R57.0	CS	73
I20.9	ACS	554	I50.2	AHF	51
I71.0	AAD	223	I44.2	TAB	50
I10.1	HE	128			

Code	Disease	Count			
I48.0	AF	769	I26.0	APE	19
I21.9	AMI	533	I51.4	AM	15
I33.0	IE	153	I47.2	VT	21
I50.1	ALHF	161	R57.0	CS	23
I20.9	ACS	170	I50.2	AHF	16
I71.0	AAD	59	I44.2	TAB	12
I10.1	HE	44			

Figure 2. Flowchart of datasets from patients in cardiovascular department to pretrain dataset and ICD triage dataset.

Table 1. Overview of target ICD codes and diseases names

Code	Disease	Abbreviation
I48.0	Atrial fibrillation	AF
I21.9	Acute myocardial infarction	AMI
I33.0	Infective endocarditis	IE
I50.1	Acute left heart failure	ALHF
I20.9	Acute coronary syndrome	ACS
I71.0	Acute aortic dissection	AAD
I10.1	Hypertensive emergency	HE
I26.0	Acute pulmonary embolism	APE
I51.4	Acute myocarditis	AM
I47.2	Ventricular tachycardia	VT
R57.0	Cardiogenic shock	CS
I50.2	Acute heart failure	AHF
I44.2	Third-degree atrioventricular block	TAB

To ensure task independence and avoid data leakage, all clinical notes were split into two parts: the pre-training corpus dataset and the ICD coding dataset. The former comprised a total of 575,632 notes, while the latter included 9,337 discharge records. The data were stratified by imbalanced ICD labels and randomly split into the training set, validation set, and test set in a 3:1:1 ratio. The sample sizes were as follows: 5,734 in the training set, 1,913 in the validation set, and 2,007 in the test set.

As shown in Figure 3, the distribution of the 13 ICD codes was imbalanced and displayed a long-tail pattern. The dataset for ICD classification includes 4.574 × 10^7 words in total, with an average of 490 words per note. The maximum and minimum lengths of clinical notes are 5,243 and 22 words, respectively.

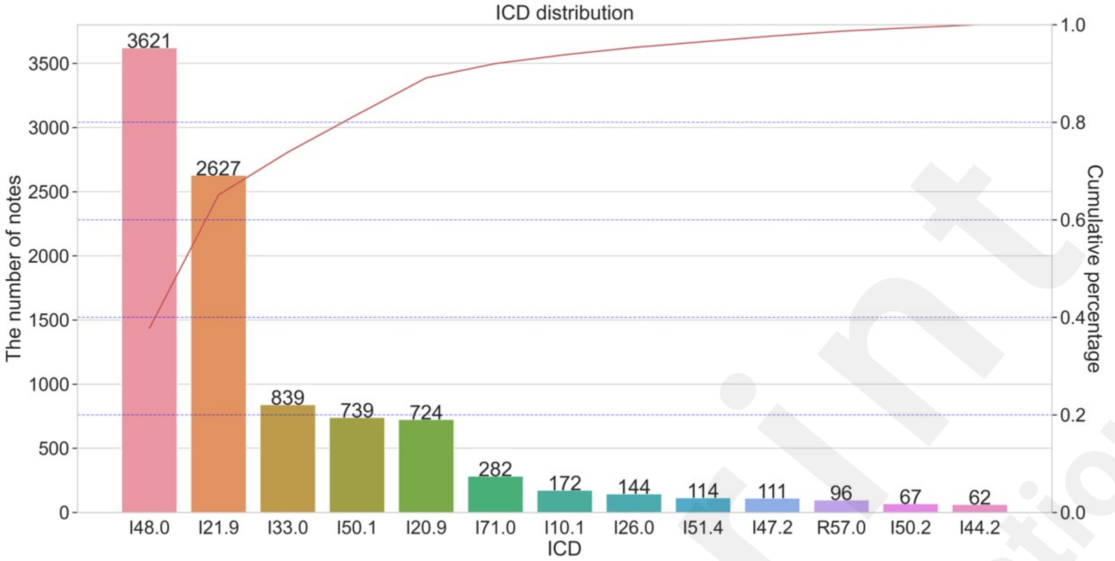


Figure 3. Distribution of ICD codes for triage task

Pretrain

Our study's foundational framework is anchored in BERT, a multi-layer bidirectional Transformer encoder renowned for its conceptual simplicity and empirical efficacy [33]. This architecture consists of 12 layers, a hidden size dimension of 768, and 12 self-attention heads [34]. BERT's inherent self-attention mechanism allows it with the versatility to accommodate diverse downstream tasks through the interchange of relevant inputs and outputs, making it well-suited for our task involving ICD classification through clinical records.

To adapt BERT to the specific demands of our task, we continued training the PLM using an extensive medical corpus, resulting in a model named MDR-BERT. During the tuning process, we selected a batch size of 32, considering the constraints of a sequence length limited to 512 tokens. The optimization algorithm Adam was employed with a conservative learning rate of 2e-5. Training iterations were conducted over a span of 15 epochs, an empirically determined figure based on the clinical dataset's characteristics.

Key-BERT

The Key-BERT method presents a novel self-supervised framework for extracting keywords and keyphrases from textual content using deep learning techniques [35]. This approach capitalizes on the inherent capabilities of contextual and semantic features derived from bidirectional transformers, with specific emphasis on the influential BERT model. The method's architecture is engineered for end-to-end training, leveraging a contextually self-annotated corpus, which imbues the model with a sophisticated comprehension of the intricate interplay between words and their semantic nuances.

A distinctive hallmark of Key-BERT resides in its automated keyword labeling process. This process ingeniously harnesses contextual insights provided by bidirectional transformers, coordinating the construction of a meticulously curated ground truth dataset. This approach circumvents the laborintensive burden of manual labeling and eliminates the prerequisite of domain-specific expertise.

The repository of self-labeled data generated by Key-BERT is partly shared with the NLP community, contributing to a broader and more profound comprehension of keyword extraction techniques across various domains. This collaborative endeavor enriches the landscape of knowledge and expertise, fostering advancements in the field of NLP and semantic information extraction.

To extract keywords using Key-BERT, the contextual feature vector of each word in a sentence is obtained by feeding the sentence into the pre-trained BERT model. Let $S = [w_1, w_2, ..., w_n]$ be a sentence consisting of *n* words, where w_i is the *i*-th word in the sentence. E_i is the contextual feature vector of the *i*-th word in the sentence. The sentence embedding vector, denoted as E_s, is obtained by averaging the feature vectors of all the words in the sentence:

$$E_i = BERT_{Embedding}(w_i) \quad (1)$$

$$E_s = \frac{E_1 + E_2 + \ldots + E_n}{n} \quad (2)$$

The cosine similarity metric is used to calculate the similarity between the sentence embedding vector and the feature vectors of candidate keywords or keyphrases.

$$\cos\text{¿}(E_i, E_s) = \frac{E_i \times E_s}{\text{¿} \vee E_i \vee \text{¿} \times \vee \text{¿} E_s \vee \text{¿}} \quad (3)$$

The top-scoring keywords or keyphrases are then returned as the most relevant to the document. Additionally, key medical terms are directly extracted based on the medical diagnostic table, ensuring that critical terminology is accurately identified and utilized.

Fine-tuning and Prompt Learning

To comprehensively harness the wealth of clinical knowledge encapsulated within the clinical dataset, our approach to fine-tuning revolves around mirroring the unsupervised task that underpins the initial pre-training phase, namely Masked Language Modeling (MLM). MLM involves the stochastic masking of a predefined proportion of input tokens. Subsequently, the model endeavors to predict these masked tokens by contextual inference, emphasizing discerning the appropriate terms within the given context. This task, often referred to as a Cloze task, facilitates the model's understanding of contextual relationships.

For the fine-tuning phase in this study, we preserved the MLM framework to ensure consistency with the pre-training procedure. A uniform masking rate of 15% was applied across the dataset. In addition to the fine-tuning process, we introduced the concept of prompt learning during parameter tuning. As part of this endeavor, the template construction incorporated four distinctive components: the input text, a soft prompt, a manual prompt, and a masking component. The manual prompt encompassed discrete tokens that mirror the downstream task anticipated by the PLM. On the contrary, the soft prompt consisted of trainable continuous vectors, enhancing the model's adaptability.

Upon the formulation of these templates, the model inputs, in conjunction with the established templates, traverse through the trainable MDR-BERT model. Notably, within the last layer of the most refined pipeline, a soft verbalizer mode was adopted. This mode orchestrates the mapping process between the predicted tokens and the final ICD codes. The innovative aspect of the soft verbalizer lies in its replacement of tokens in the verbalizer with vectors that are amenable to training, each tailored to a specific class. This strategy enhances the precision and semantic fidelity of the generated outputs, facilitating a more refined correspondence between predicted tokens and definitive ICD codes.

Consequently, it is unnecessary to manually build an explicit mapping $g : V \mapsto Y$ for the soft verbalizer, as the trainable vectors do not have explainable semantic meaning. A matrix operator can represent the soft verbalizer as $\Theta \in R^{n \times m}$ [20-23], where *n* represents the size of Y and *m* represents the dimension of output embeddings from M. For the verbalizer, θ_i denotes the *i*th row of Θ as the trainable vector of the *i*th class. The soft verbalizer replaces the original decoder head of M by mapping the embeddings of x' from M, denoted as $e(x')$, to the distribution over the classes of Y. We denote the resulting mapping from $e(x') \in R^{l \times m}$ to the prediction of the embedding of <[MASK]> as $f_{mask} : R^{l \times m} \mapsto R^m$, where *l* is the sequence length of x'. And then, the probability of class *y* can be calculated as

$$P(y|x) = \frac{\exp\left(\theta_y^{\top} f_{mask}(e(x'))\right)}{\sum_{i=1}^{n} \exp\left(\theta_i^{\top} f_{mask}(e(x'))\right)} \tag{4}$$

The loss from the automatic ICD coding task can then be backpropagated to tune only the embeddings for the prompt template and verbalizer. The loss function could be expressed as:

$$\hat{y} = argmax_{y \in Y} \text{¿} \tag{5}$$

Finally, the model learns to generate and map the most appropriate codes to the corresponding discharge record.

The experiments were implemented based on the OpenPrompt framework [20-23]. In prompt learning, we adopted the Adafactor optimizer for soft and mixed templates of prompts and the AdamW optimizer for PLMs and soft verbalizers. In conventional fine-tuning, we used the AdamW optimizer for MLP heads and PLMs. We accelerated the experiments using two Nvidia TESLA V100 GPUs with 16GB memory each, and the batch size was set to 32 due to memory limitations.

The performance of the model varies with changes to hyperparameters. In the following comparisons, hyperparameters were carefully optimized for each model. A random search strategy was performed to derive optimal hyperparameters for the training runs. This strategy consisted of 100 training runs using randomly generated hyperparameters from the corresponding search space. The optimal hyperparameters for the models are shown in Table 2.

Table 2. The optimal hyperparameters and their search space

Hyperparameters	Search space	Optimal hyperparameter	
		Prompt learn	Fine-tune
learning rate	log.uniform [1*10-5, 3*10-1]	0.0048	0.0121
batch size	[4]	4	4
gradient accumulation steps	range[2,10]	4	3
dropout	range[0.1,0.5]	0.382	0.1563
optimizer	[adamw, adafactor]	adamw	adafactor
prompt learning rate	log.uniform[1*10-5, 3*10-1]	0.3	-
verbalizer learning rate	log.uniform[1*10-5, 1*10-1]	0.007	-

Evaluation Metrics

To evaluate the performance of models for comprehensive comparison, we adopted a variety of metrics, including micro-F1, macro-AUC (the area under the ROC curve), and accuracy. The micro-averaged precision and F1 are defined in Eqs. 6–8, and the macro-AUC is defined in Eqs. 9-10.

$$Micro-P=\frac{\sum_{i=1}^{I} TP_i}{\sum_{i=1}^{I} TP_i+FP_i} \tag{6}$$

$$Micro-R=\frac{\sum_{i=1}^{I} TP_i}{\sum_{i=1}^{I} TP_i+FN_i} \tag{7}$$

$$Micro-F1=\frac{2*(Micro-P)*(Micro-R)}{(Micro-P)+(Micro-R)} \tag{8}$$

where TP_i, FP_i, FN_i represent true positives (correctly assigned instances), false positives (incorrect assignments by automated methods), and false negatives (correct instances omitted by automated methods), respectively, of code *i*, and *l* is the size of the sample space. The $Micro-F1$ is the harmonic mean of $Micro-P$ and $Micro-R$, and a bigger value of $Micro-F1$ indicates a better performance.

$$\mathrm{AUC}_k=\frac{1}{2}\sum_{i=1}^{n-1}\left(\mathrm{TPR}_k(i+1)+\mathrm{TPR}_k(i)\right)\left(\mathrm{FPR}_k(i+1)-\mathrm{FPR}_k(i)\right) \tag{9}$$

$$\text{Macro - AUC}=\frac{1}{K}\sum_{k=1}^{K}\mathrm{AUC}_k \tag{10}$$

where n is the number of thresholds and K is the number of classes.

Data and Code availability

Data acquisition can be requested by contacting the provided email address. Due to the sensitivity of the hospital data, it cannot be made publicly available. Part of the downstream subtask data is currently undergoing desensitization and approval processes. The source code is publicly available at https://github.com/PLA301dbgroup2/ICD_promptLearning.

Results

Performance of Different Pipelines

To evaluate the performance of different methods, we applied four state-of-the-art algorithms: BERT [13], XLNet [16], RoBERTa [17, 31], and prompt learning [20]. We combined these PLMs with various algorithms to create six main pipelines: BERT with fine-tuning, XLNet with fine-tuning, RoBERTa with fine-tuning, BERT with prompt learning, MDR-BERT with prompt learning, and MDR-BERT with fine-tuning and prompt learning. MDR-BERT is a PLM obtained by further pre-training BERT on our medical corpus.

As shown in Figure 4, MDR-BERT with fine-tuning and prompt learning achieved the best results across all evaluation metrics, with a micro-F1 score of 0.838, a macro-AUC of 0.958, and an accuracy of 0.838. MDR-BERT with prompt learning alone performed slightly worse than MDR-BERT with fine-tuning and prompt learning, but both outperformed the other pipelines by a large margin. This suggests that continued pre-training on clinical records can significantly improve the PLM's performance on the task and that freezing the parameters may hinder the adaptation of small-sized PLMs to the task.

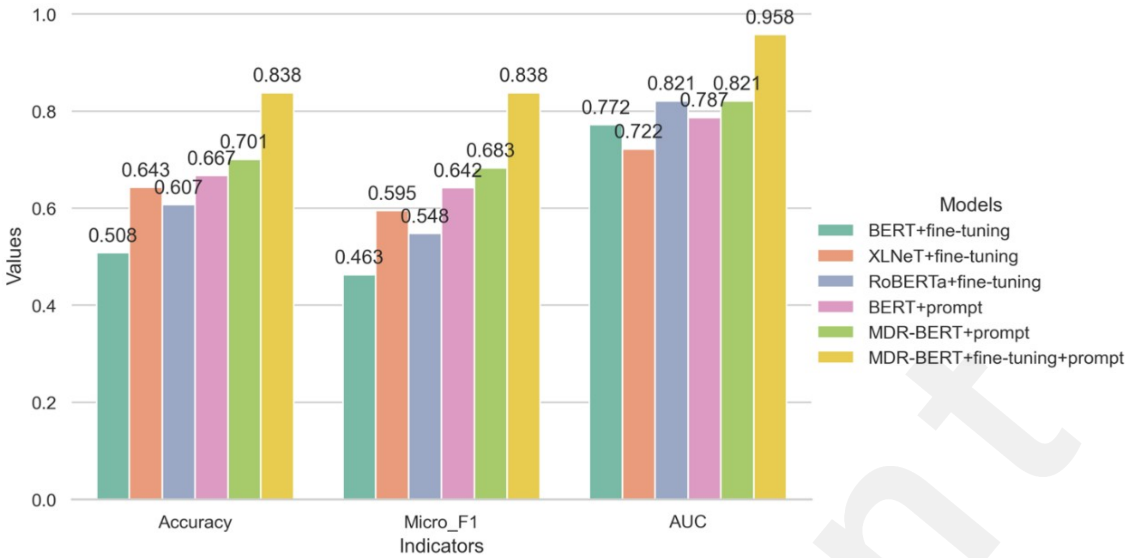


Figure 4. Comparison among different pipelines in accuracy, micro f1, and AUC.

Among the other pipelines, BERT with prompt learning had the highest accuracy (0.67) and the highest micro-F1 score (0.64), although its macro-AUC (0.79) was slightly lower than that of RoBERTa with fine-tuning. This indicates that prompt learning, as a lightweight tuning method, can match or even surpass traditional fine-tuning methods, which is consistent with the findings of Taylor et al. [20].

Performance of Different Prompt Learning Mode

We evaluated the performance of MDR-BERT under various settings of prompt learning and fine-tuning, utilizing three types of templates (manual, soft, and mixed) and two types of verbalizers (manual and soft) as hyper-parameters. As shown in Figure 5, the combination of mixed templates and the soft verbalizer achieved the best results.

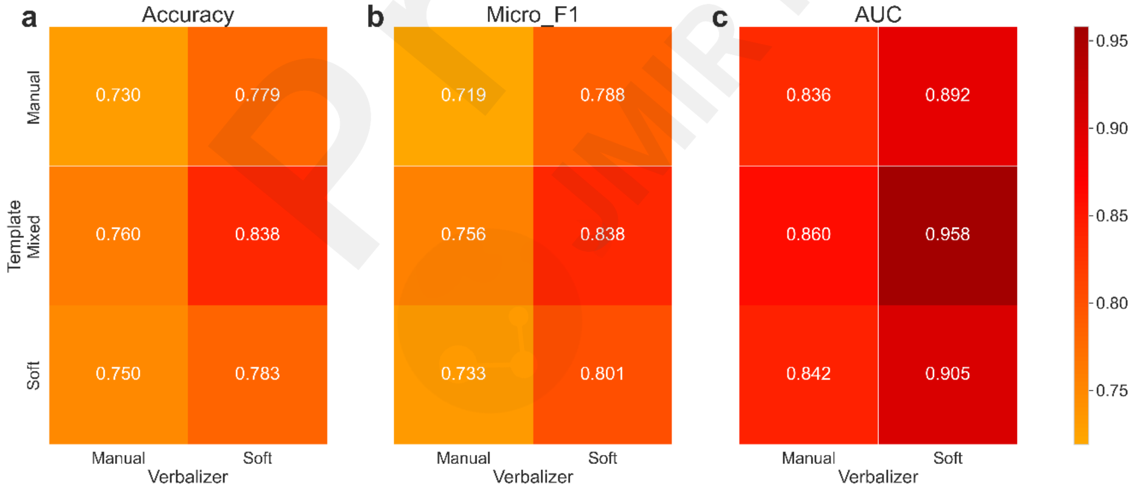


Figure 5. Comparison among different prompt combinations in verbalizer and template.

For templates, both scripted and self-adaptive patterns performed well independently, and their combination had a cumulative positive effect on performance. For verbalizers, the self-adaptive type significantly outperformed the traditional manual vectors, having a greater impact on the overall performance.

Performance of MDR-BERT with Fine-tuning and Prompt Learning

We evaluated the performance of the MDR-BERT pipeline, incorporating both fine-tuning and prompt learning, for each ICD code using precision, recall, and F1 score. Figure 6 presents the results for these metrics across the 13 ICD classes.

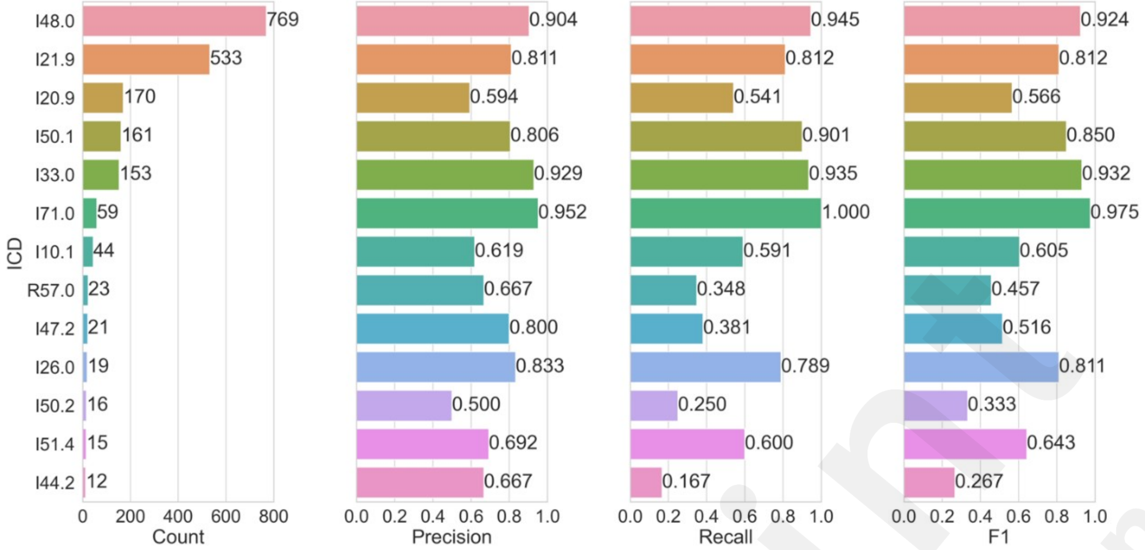


Figure 6. Precision, recall, and f1 scores of every ICD code in pipeline of MDR-BERT with fine-tuning and prompt learning

The pipeline achieved high scores for most ICD codes, though the scores varied depending on the data distribution and sample size for each code. We observed a weak positive correlation between sample size and model performance, suggesting that larger samples enhanced the model's learning capability. Conversely, smaller samples tended to have lower F1 scores, with a trade-off between precision and recall for certain classes. Despite these variations, our pipeline demonstrated satisfactory performance levels across the different ICD codes.

Few-shot Learning

We conducted few-shot experiments to evaluate the performance of the fine-tuned MDR-BERT with the prompt learning pipeline using different sample sizes from the training set. We randomly selected samples ranging from 1 to 4000 and evaluated the models on the test set. Figure 7 shows the accuracy, micro-F1, and macro-AUC scores for each sample size.

The results indicated that the pipeline performed poorly in small-scale few-shot scenarios. However, the scores increased rapidly as the sample size grew, reaching a plateau at around 500 samples. This suggests that the PLM relies heavily on sample size and requires a minimum amount of data to achieve consistent performance.

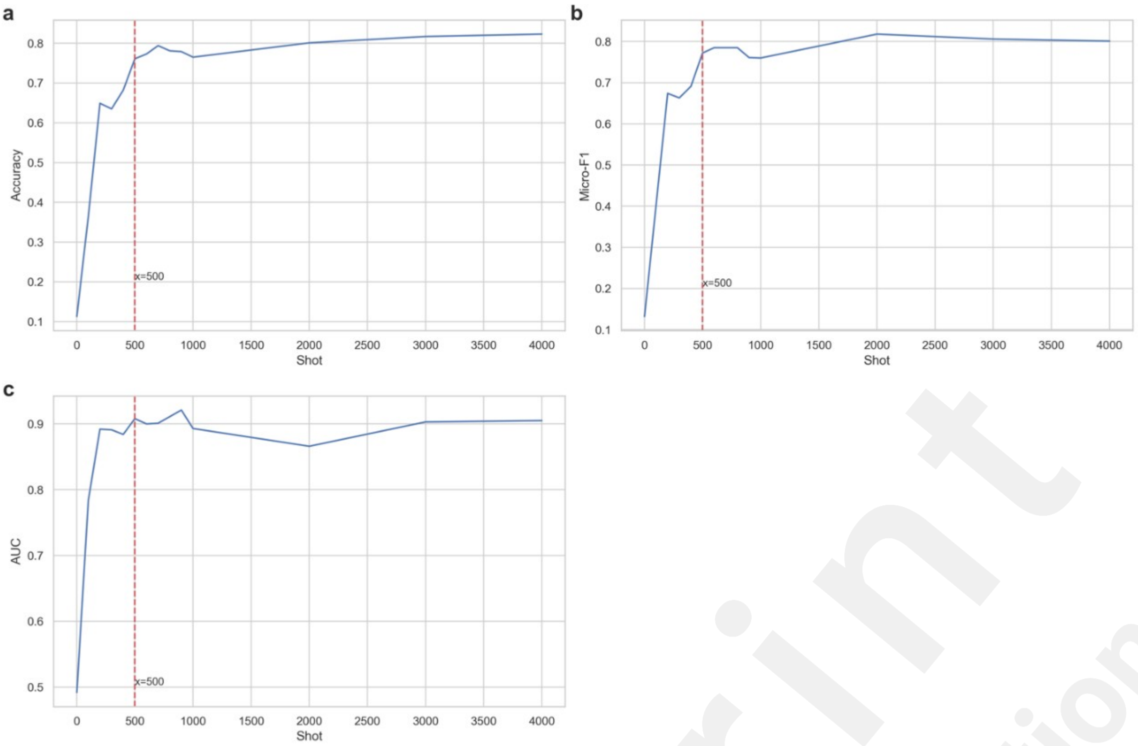


Figure 7. Few shots experiments on MDR-BERT with fine-tuning and prompt learning

Discussion

Principle Results

An automated ICD coding system for long free texts is a fundamental platform for clinical research and practice, including clinical trials and pharmacoeconomic management. In this study, we developed a framework based on key-BERT, a continuously trained and tunable PLM, and task-specific prompt learning. We collected a total of 584,969 clinical notes from admission records and discharge summaries in the cardiovascular departments of eight medical centers.

We utilized most of the data to continue pre-training a medical corpus, named MDR-BERT, and used an independent set of 9,337 discharge records with 13 ICD codes of CVDs for the ICD classification subtask. To remove irrelevant information and restrict the input token size, we filtered and truncated all the data for the ICD task into pieces of keywords using key-BERT. The data was then stratified and split into training, validation, and test sets. The test set was used independently for the final evaluation.

This study primarily focused on transformer-based algorithms, which have been widely applied and demonstrated superior performance in large-scale medical long free text tasks [4, 9, 14, 15]. These algorithms can leverage PLMs that capture the semantic and syntactic information of natural language from extensive corpora, significantly benefiting performance improvement through multi-center datasets.

We compared six pipelines for the classification downstream task: BERT with fine-tuning, XLNet with fine-tuning, RoBERTa with fine-tuning, frozen BERT with prompt learning, frozen MDR-BERT with prompt learning, and tunable MDR-BERT with prompt learning. The prompt learning setup included three types of templates and two types of verbalizers. Among the pipelines, MDR-BERT with fine-tuning and prompt learning achieved the best performance on the test set, attaining a micro-

F1 score of 0.838, a macro-AUC of 0.958, and an accuracy of 0.838.

The favorable outcome of this pipeline can be attributed to the use of a large-scale corpus-based PLM and the task-specific improvements from the combination of fine-tuning and prompt learning [14, 20-23, 32]. Fine-tuning serves as a model adapter, aligning the model distribution with the task distribution, thereby overcoming domain shift and task mismatch issues inherent in PLMs. Prompt learning, with its compact prefix representation and sparse attention mechanism, augments the training data with diverse and natural examples. This augmentation helps alleviate data scarcity and label noise issues in small-sized datasets for downstream tasks.

The combination of fine-tuning and prompt learning acts as a regularization term that balances model complexity with data quality, ultimately enhancing overall performance. This integrated approach demonstrates the potential of leveraging advanced transformer-based models and tailored learning strategies to improve automated medical coding and other clinical tasks.

Among the different prompt learning setups, the mixed template and soft verbalizer achieved the best performance. The soft template method outperformed the manual templates method, which may be attributed to the greater semantic and syntactic information, broader search space, and reduced trial-and-error process of the soft template method, making it more effective and less time-consuming [21, 22].

The mixed template method is a hybrid approach that combines the advantages of both soft and manual templates. It uses a manual template as a base prompt to provide human-readable instructions and natural language labels, and a soft template as an auxiliary prompt to provide tunable embeddings that can adapt to specific downstream tasks. Hence, the manual template leverages existing knowledge, while the soft template enhances expressiveness and flexibility.

For the verbalizer, the self-adaptive type had a significantly higher impact on overall performance than traditional manual vectors. The soft verbalizer adjusts to the optimal label space for each task and the scale of the pre-trained model, rather than being constrained by a fixed set of tokens [20, 22]. This improves the accuracy and robustness of the predictions, as well as the diversity and naturalness of the labels. Moreover, by tuning the verbalizer along with other continuous prompts, it retains the advantages of prompt tuning over fine-tuning, avoiding the need to keep a copy of model parameters for each task at inference.

To explore the influence of sample size on the performance of our pipeline, we conducted few-shot experiments with a range of 1 to 4000 shots. The results indicated unsatisfactory evaluation metrics in small-scale shots, but the performance improved rapidly and stabilized at around 500 shots. This suggests that for mid-sized language models, such as BERT, the semantic understanding and representation capabilities may not be sufficiently strong. Therefore, tuning the parameters of the PLM with a certain sample size is necessary to achieve better performance on specific tasks.

Limitations

Despite the reasonable performance of our pipeline, this study has certain limitations. Firstly, we only trained both the corpus part and the classification task of the framework in the cardiovascular department. As a result, the conclusions of this paper may not be generalizable to other medical fields. Secondly, the subtask of ICD classification only involved ICD codes of 13 CVDs, which is not comprehensive enough for clinical practice. Future research could extend to explore the automatic encoding of additional critical heart diseases or even the entire clinical field. This could potentially broaden the applicability and effectiveness of the proposed approach in addressing a wider range of clinical tasks.

Conclusion

We proposed a real-time framework for ICD coding from medical long free text to ICD labels, without the need for semi-structured preprocessing. This framework consists of key-BERT, a continuously trained and tunable PLM, and task-specific prompt learning with mixed templates and soft verbalizers. We evaluated our model on a multi-center cardiovascular dataset and applied it to predict 13 ICD codes for CVDs, achieving high performance. Our model also demonstrated transferability and generalization across different centers.

Furthermore, we conducted few-shot experiments to investigate the impact of data size on model performance. The results showed that while the framework was effective on smaller datasets, it required a certain amount of sample size to achieve a relatively stable performance level. This study serves as a benchmark for exploring the feasibility and performance of prompt learning in the subtask of large language models or PLMs. Utilizing a multi-center dataset, the approach demonstrated robust performance across hospitals, highlighting its potential for broad deployment.

Few-shot learning experiments exhibited feasibility in small-scale datasets, enabling applications for local training on single centers or various single-disease databases. The real-time model identifies ICD codes directly, accelerating automated coding compared to semi-automatic approaches with segment preprocessing. This is particularly impactful for clinical decision support systems relying on real-time ICD coding data.

Overall, the prompt learning paradigm achieved cutting-edge ICD assignment accuracy while offering deployability, few-shot learning capacity, and low latency, which are beneficial for healthcare applications. This automated ICD coding pipeline could be further implemented for various clinical applications, such as clinical decision support systems, cohort studies, and disease early warning and diagnosis systems.

Ethics approval

The study was approved by the Institutional Ethics Committee of Beijing Natural Science Foundation (S2023-324-02).

Acknowledgments

This work was supported by Beijing Natural Science Foundation (L222006), National Key R&D Program of China (2021ZD0140406, 2021ZD0140408), National Natural Science Foundation of China (62072261), and Independent Research Project of Medical Engineering Laboratory of Chinese PLA General Hospital (2022SYSZZKY23).

Conflicts of interests

The authors declare no competing interests.

Abbreviations

BERT: Bidirectional Encoder Representations from Transformers
CVD: Cardiovascular Disease
EHR: Electronic Health Record
ICD: International Classification of Diseases
NLP: Natural Language Processing

PLM: Pre-trained Language Models

References

1. Steindel, S.J. (2010). International classification of diseases, 10th edition, clinical modification and procedure coding system: descriptive overview of the next generation HIPAA code sets. J Am Med Inform Assoc 17, 274-282. 10.1136/jamia.2009.001230.

2. O'Malley, K.J., Cook, K.F., Price, M.D., Wildes, K.R., Hurdle, J.F., and M., A.C. (2005). Measuring Diagnoses: ICD Code Accuracy. Health Services Research 40, 1620-1639.

3. Kusnoor, S.V., Blasingame, M.N., Williams, A.M., DesAutels, S.J., Su, J., and Giuse, N.B. (2020). A narrative review of the impact of the transition to ICD-10 and ICD-10-CM/PCS. JAMIA Open 3, 126-131. 10.1093/jamiaopen/ooz066.

4. Chen, P.F., Chen, K.C., Liao, W.C., Lai, F., He, T.L., Lin, S.C., Chen, W.J., Yang, C.Y., Lin, Y.C., Tsai, I.C., et al. (2022). Automatic International Classification of Diseases Coding System: Deep Contextualized Language Model With Rule-Based Approaches. JMIR Med Inform 10, e37557. 10.2196/37557.

5. Upadhyaya, S.G., Murphree, D.H., Jr., Ngufor, C.G., Knight, A.M., Cronk, D.J., Cima, R.R., Curry, T.B., Pathak, J., Carter, R.E., and Kor, D.J. (2017). Automated Diabetes Case Identification Using Electronic Health Record Data at a Tertiary Care Facility. Mayo Clin Proc Innov Qual Outcomes 1, 100-110. 10.1016/j.mayocpiqo.2017.04.005.

6. Diao, X., Huo, Y., Zhao, S., Yuan, J., Cui, M., Wang, Y., Lian, X., and Zhao, W. (2021). Automated ICD coding for primary diagnosis via clinically interpretable machine learning. Int J Med Inform 153, 104543. 10.1016/j.ijmedinf.2021.104543.

7. Maheshwari, S., Agarwal, A., Shukla, A., and Tiwari, R. (2020). A comprehensive evaluation for the prediction of mortality in intensive care units with LSTM networks: patients with cardiovascular disease. Biomed Tech (Berl) 65, 435-446. 10.1515/bmt-2018-0206.

8. Bao, W., Lin, H., Zhang, Y., Wang, J., and Zhang, S. (2021). Medical code prediction via capsule networks and ICD knowledge. BMC Med Inform Decis Mak 21, 55. 10.1186/s12911-021-01426-9.

9. Kreuzthaler, M., Pfeifer, B., Kramer, D., and Schulz, S. (2023). Secondary Use of Clinical Problem List Entries for Neural Network-Based Disease Code Assignment. Stud Health Technol Inform 302, 788-792. 10.3233/SHTI230267.

10. Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep Contextualized Word Representations. arXiv:1802.05365.

11. Dong, L., Yang, N., Wang, W., Wei, F., Liu, X., Wang, Y., Gao, J., Zhou, M., and Hon, H.-W. (2019). Unified Language Model Pre-training for Natural Language Understanding and Generation. arXiv:1905.03197.

12. Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. (2020). BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. arXiv:1910.13461.

13. Devlin, J., Chang, M.W., Lee, K., and Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Preprints. 1810.04805v2.

14. Coutinho, I., and Martins, B. (2022). Transformer-based models for ICD-10 coding of death certificates with Portuguese text. J Biomed Inform 136, 104232. 10.1016/j.jbi.2022.104232.

15. Yan, A., McAuley, J., Lu, X., Du, J., Chang, E.Y., Gentili, A., and Hsu, C.N. (2022). RadBERT: Adapting Transformer-based Language Models to Radiology. Radiol Artif Intell 4, e210258. 10.1148/ryai.210258.

16. Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., and Le, Q.V. (2019). XLNet: Generalized Autoregressive Pretraining for Language Understanding. Preprints. 10.48550/arXiv.1906.08237.

17. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. Preprints. 10.48550/arXiv.1907.11692.

18. Liu, X., Zheng, Y., Du, Z., Ding, M., Qian, Y., Yang, Z., and Tang, J. (2021). GPT understands, too. arXiv preprint arXiv:2103.10385.

19. Schick, T., and Schütze, H. (2021). Exploiting cloze questions for few shot text classification and natural language inference. European Chapter of the Association for Computational Linguistics. arXiv:2001.07676.

20. Taylor, N., Zhang, Y., Joyce, D., Nevado-Holgado, A., and Kormilitzin, A. (2022). Clinical Prompt Learning with Frozen Language Models. Preprints, arXiv:2205.05535. 10.48550/arXiv.2205.05535.

21. Li, X.L., and Liang, P. (2021). Prefix-Tuning: Optimizing Continuous Prompts for Generation. Preprints. 10.48550/arXiv.2101.00190.

22. Liu, X., Ji, K., Fu, Y., Du, Z., Yang, Z., and Tang, J. (2021). P-Tuning v2: Prompt Tuning Can Be Comparable to Fine-tuning Universally Across Scales and Tasks. Preprints. 10.48550/arXiv.2110.07602.

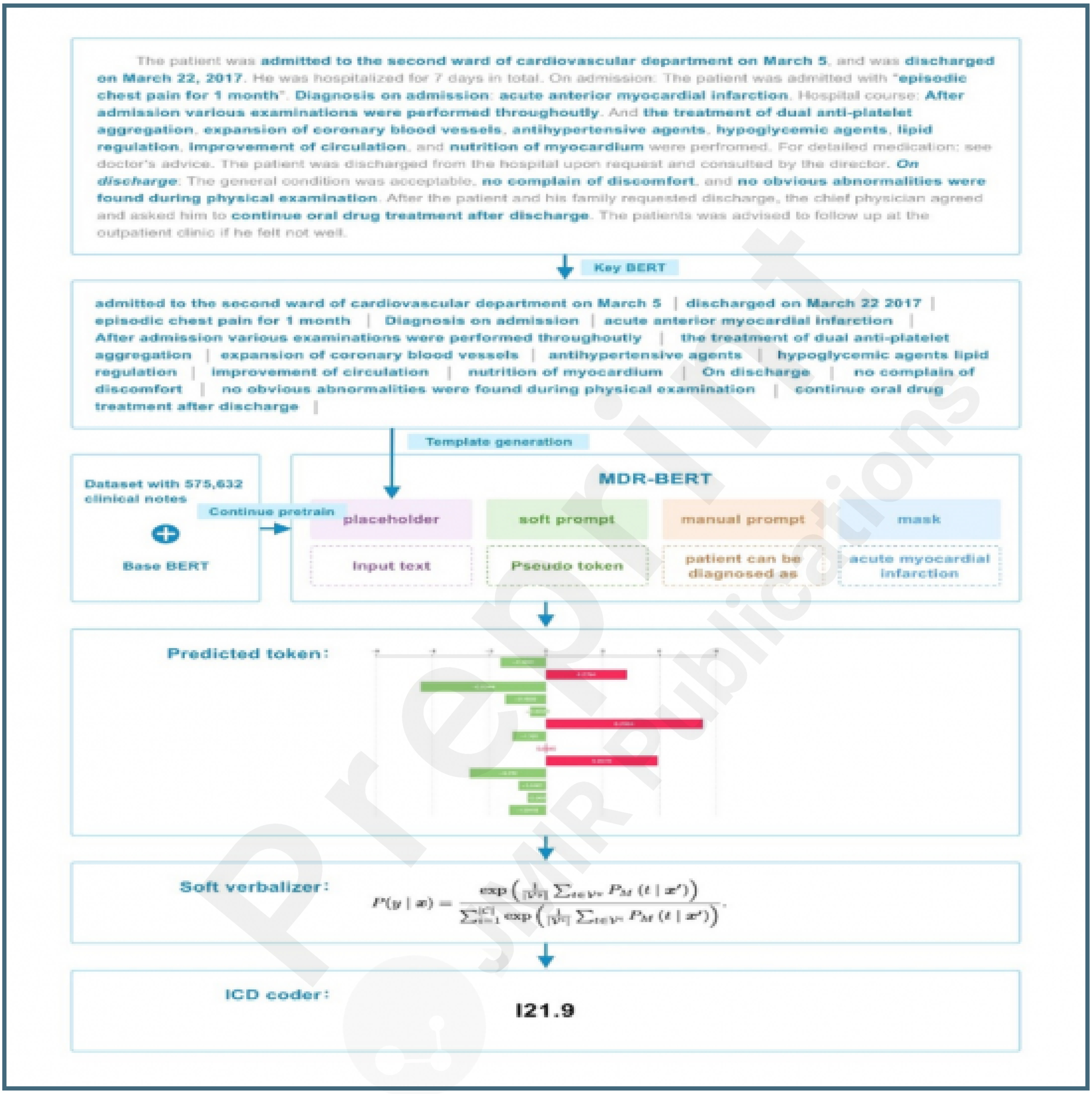
23. Ding, N., Hu, S., Zhao, W., Chen, Y., Liu, Z., Zheng, H.-T., and Sun, M. (2021). OpenPrompt: An Open-

source Framework for Prompt-learning. Preprints, arXiv:2111.01998. 10.48550/arXiv.2111.01998.
24. Jiang, Z., Xu, F.F., Araki, J., and Neubig, G. (2020). How can we know what language models know? arXiv:1911.12543.
25. Haviv, A., Berant, J., and Globerson, A. (2021). BERTese: Learning to Speak to BERT. arXiv:2103.05327.
26. Wallace, E., Feng, S., Kandpai, N., Gardner, M., and Singh, S. (2019). Universal adversarial triggers for attacking and analyzing NLP. arXiv:1908.07125.
27. Yin, W., Hay, J., and Roth, D. (2019). Template-Based Named Entity Recognition Using BART.
28. Gao, T., Fisch, A., and Chen, D. (2021). Making pre-trained language models better few-shot learners. arXiv:2012.15723.
29. Shin, R., Lin, C.H., Thomson, S., Chen, C., Roy, S., Platanios, E.A., Pauls, A., Klein, D., Eisner, J., and Durme, B.V. (2021). Constrained language models yield few-shot semantic parsers. arXiv:2104.08768.
30. Schick, T., Schmid, H., and Schutze, H. (2020). Automatically identifying words that can serve as labels for few-shot text classification. arXiv:2010.13641.
31. Tinn, R., Cheng, H., Gu, Y., Usuyama, N., Liu, X., Naumann, T., Gao, J., and Poon, H. (2023). Fine-tuning large neural language models for biomedical natural language processing. Patterns (N Y) 4, 100729. 10.1016/j.patter.2023.100729.
32. Liu, X., Zheng, Y., Du, Z., Ding, M., Qian, Y., Yang, Z., and Tang, J. (2021). GPT Understands, Too. Preprints, arXiv:2103.10385. 10.48550/arXiv.2103.10385.
33. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., and Polosukhin, I. (2017). Attention Is All You Need. arXiv:1706.03762.
34. Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805.
35. Sharma, P., and Li, Y. (2019). Self-Supervised Contextual Keyword and Keyphrase Retrieval with Self-Labelling. Preprints. 10.20944/preprints201908.0073.v1.

Supplementary Files

Figures

Overall framework of MDR-BERT, key-BERT, prompt learning pipeline.

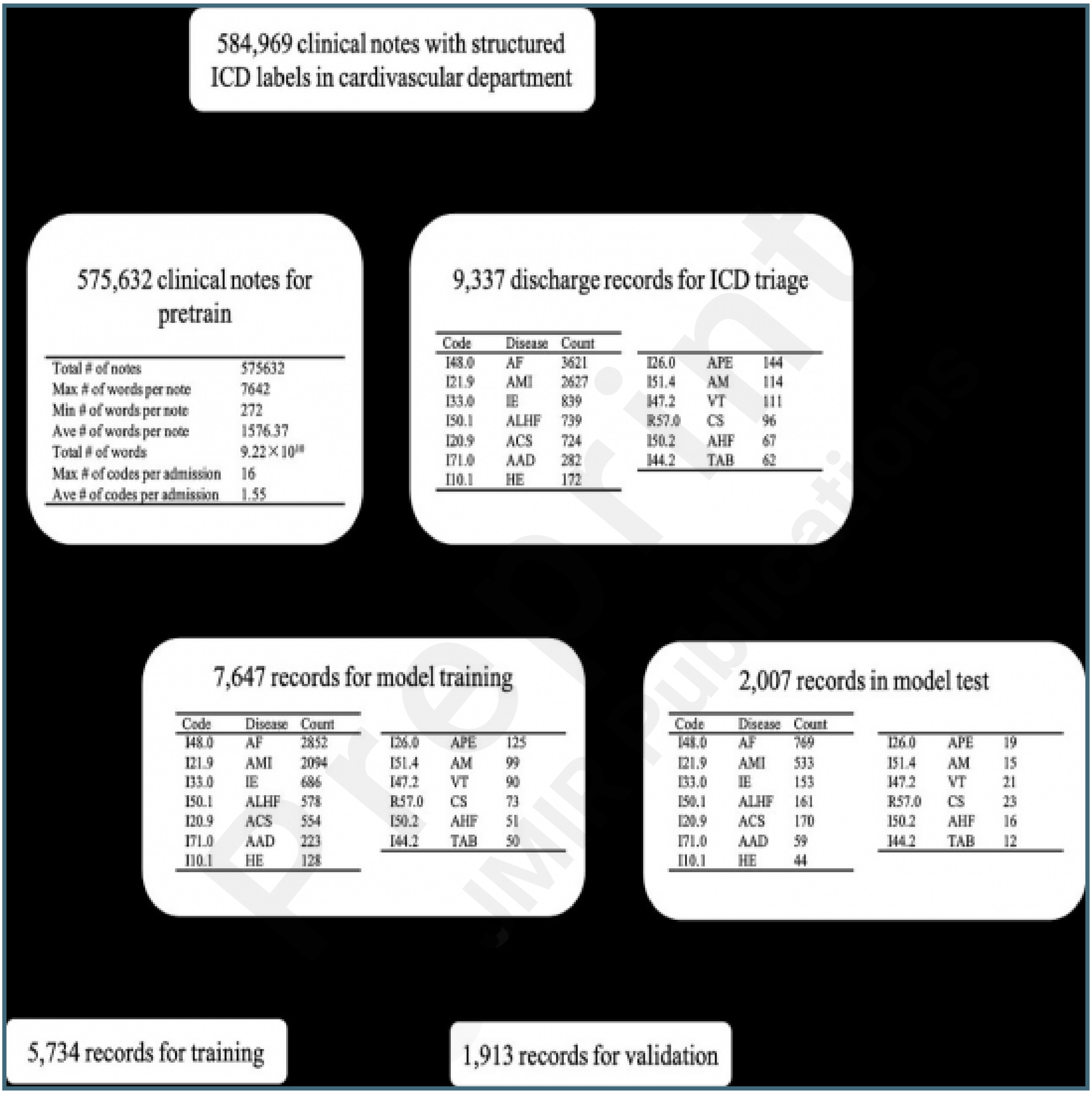


The patient was admitted to the second ward of cardiovascular department on March 5, and was discharged on March 22, 2017. He was hospitalized for 7 days in total. On admission: The patient was admitted with "episodic chest pain for 1 month". Diagnosis on admission: acute anterior myocardial infarction. Hospital course: After admission various examinations were performed throughoutly. And the treatment of dual anti-platelet aggregation, expansion of coronary blood vessels, antihypertensive agents, hypoglycemic agents, lipid regulation, improvement of circulation, and nutrition of myocardium were perfromed. For detailed medication: see doctor's advice. The patient was discharged from the hospital upon request and consulted by the director. On discharge: The general condition was acceptable, no complain of discomfort, and no obvious abnormalities were found during physical examination. After the patient and his family requested discharge, the chief physician agreed and asked him to continue oral drug treatment after discharge. The patients was advised to follow up at the outpatient clinic if he felt not well.

admitted to the second ward of cardiovascular department on March 5 | discharged on March 22 2017 | episodic chest pain for 1 month | Diagnosis on admission | acute anterior myocardial infarction | After admission various examinations were performed throughoutly | the treatment of dual anti-platelet aggregation | expansion of coronary blood vessels | antihypertensive agents | hypoglycemic agents lipid regulation | improvement of circulation | nutrition of myocardium | On discharge | no complain of discomfort | no obvious abnormalities were found during physical examination | continue oral drug treatment after discharge |

$$P(y \mid x) = \frac{\exp\left(\frac{1}{|V_y|}\sum_{t\in V_y} P_M(t \mid x')\right)}{\sum_{i=1}^{|C|}\exp\left(\frac{1}{|V_i|}\sum_{t\in V_i} P_M(t \mid x')\right)}.$$

Flowchart of datasets from patients in cardiovascular department to pretrain dataset and ICD triage dataset.


584,969 clinical notes with structured ICD labels in cardivascular department
575,632 clinical notes for pretrain
9,337 discharge records for ICD triage
7,647 records for model training
2,007 records in model test
5,734 records for training
1,913 records for validation

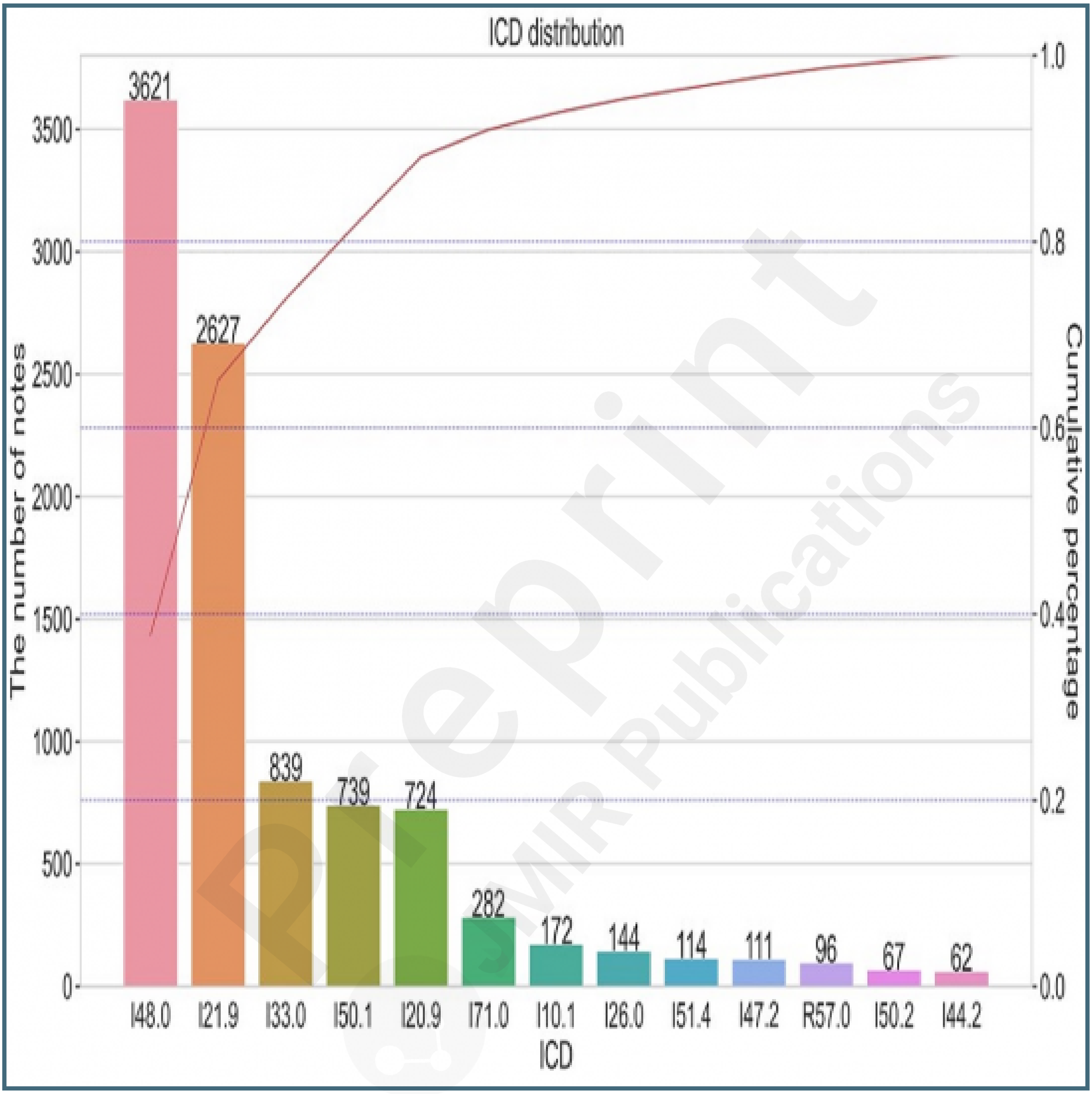
Total # of notes	575632
Max # of words per note	7642
Min # of words per note	272
Ave # of words per note	1576.37
Total # of words	9.22×10^{10}
Max # of codes per admission	16
Ave # of codes per admission	1.55

Code	Disease	Count
I48.0	AF	3621
I21.9	AMI	2627
I33.0	IE	839
I50.1	ALHF	739
I20.9	ACS	724
I71.0	AAD	282
I10.1	HE	172
I26.0	APE	144
I51.4	AM	114
I47.2	VT	111
R57.0	CS	96
I50.2	AHF	67
I44.2	TAB	62

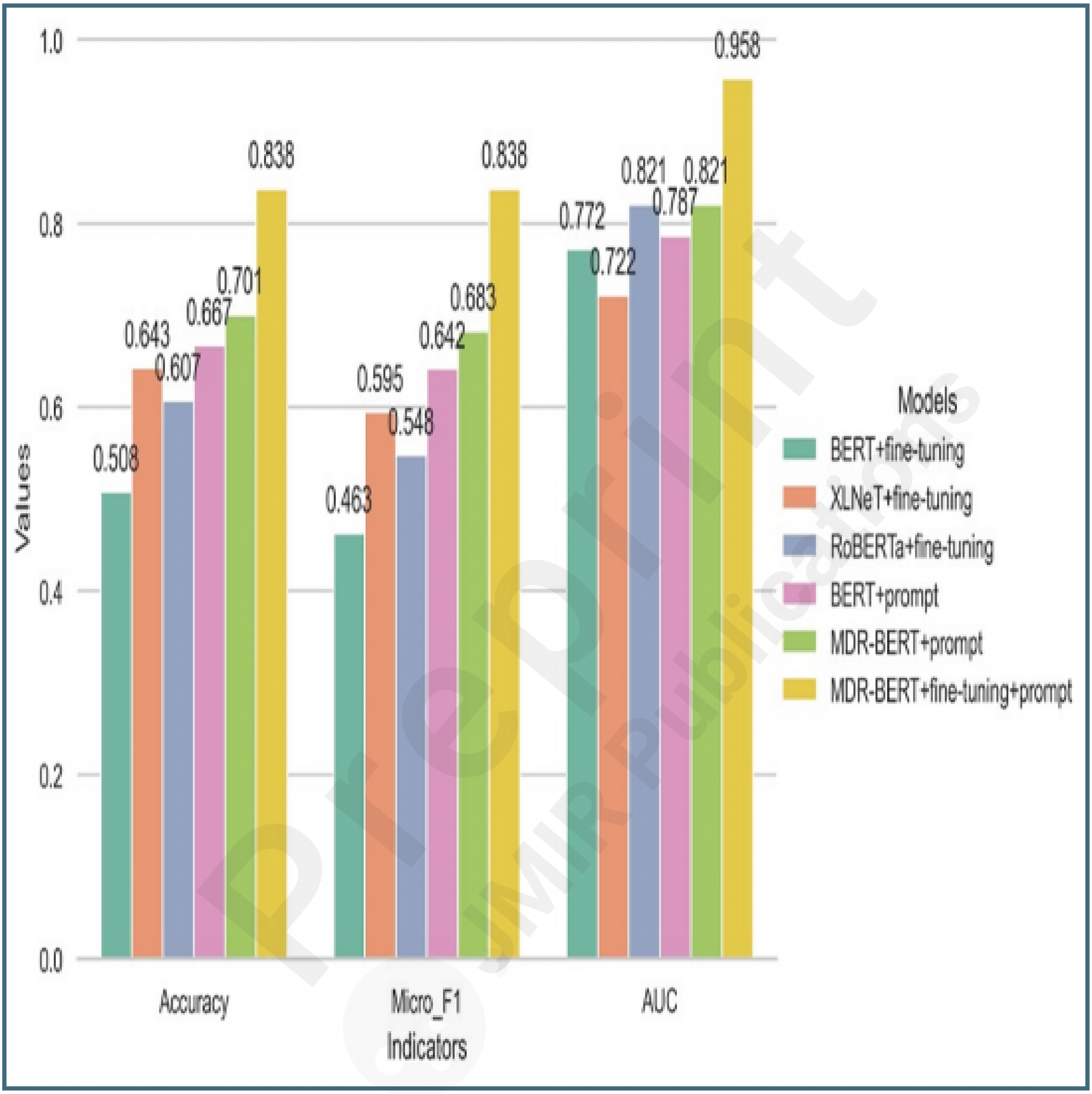
Code	Disease	Count
I48.0	AF	2852
I21.9	AMI	2094
I33.0	IE	686
I50.1	ALHF	578
I20.9	ACS	554
I71.0	AAD	223
I10.1	HE	128
I26.0	APE	125
I51.4	AM	99
I47.2	VT	90
R57.0	CS	73
I50.2	AHF	51
I44.2	TAB	50

Code	Disease	Count
I48.0	AF	769
I21.9	AMI	533
I33.0	IE	153
I50.1	ALHF	161
I20.9	ACS	170
I71.0	AAD	59
I10.1	HE	44
I26.0	APE	19
I51.4	AM	15
I47.2	VT	21
R57.0	CS	23
I50.2	AHF	16
I44.2	TAB	12

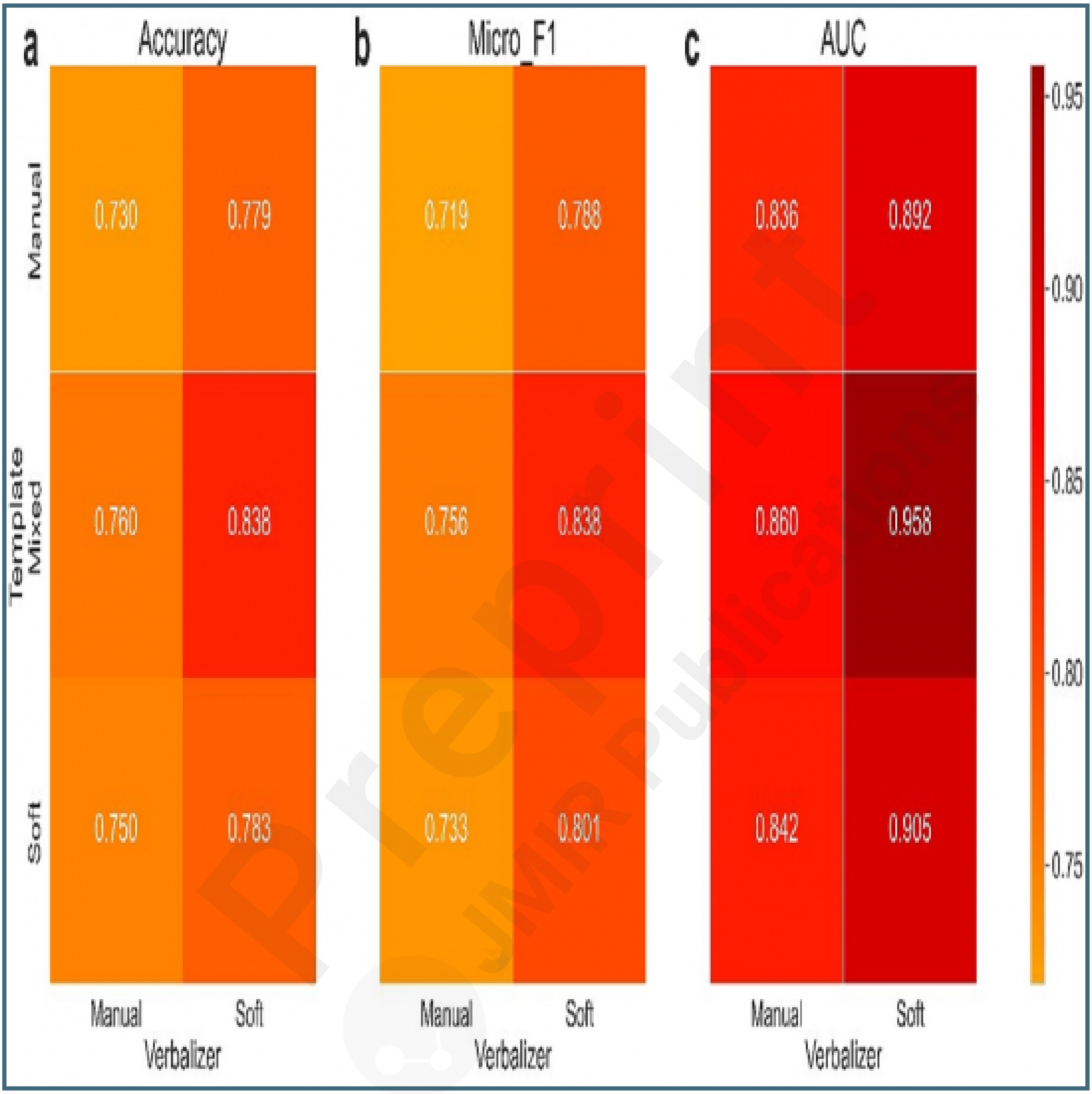
Distribution of ICD codes for triage task.



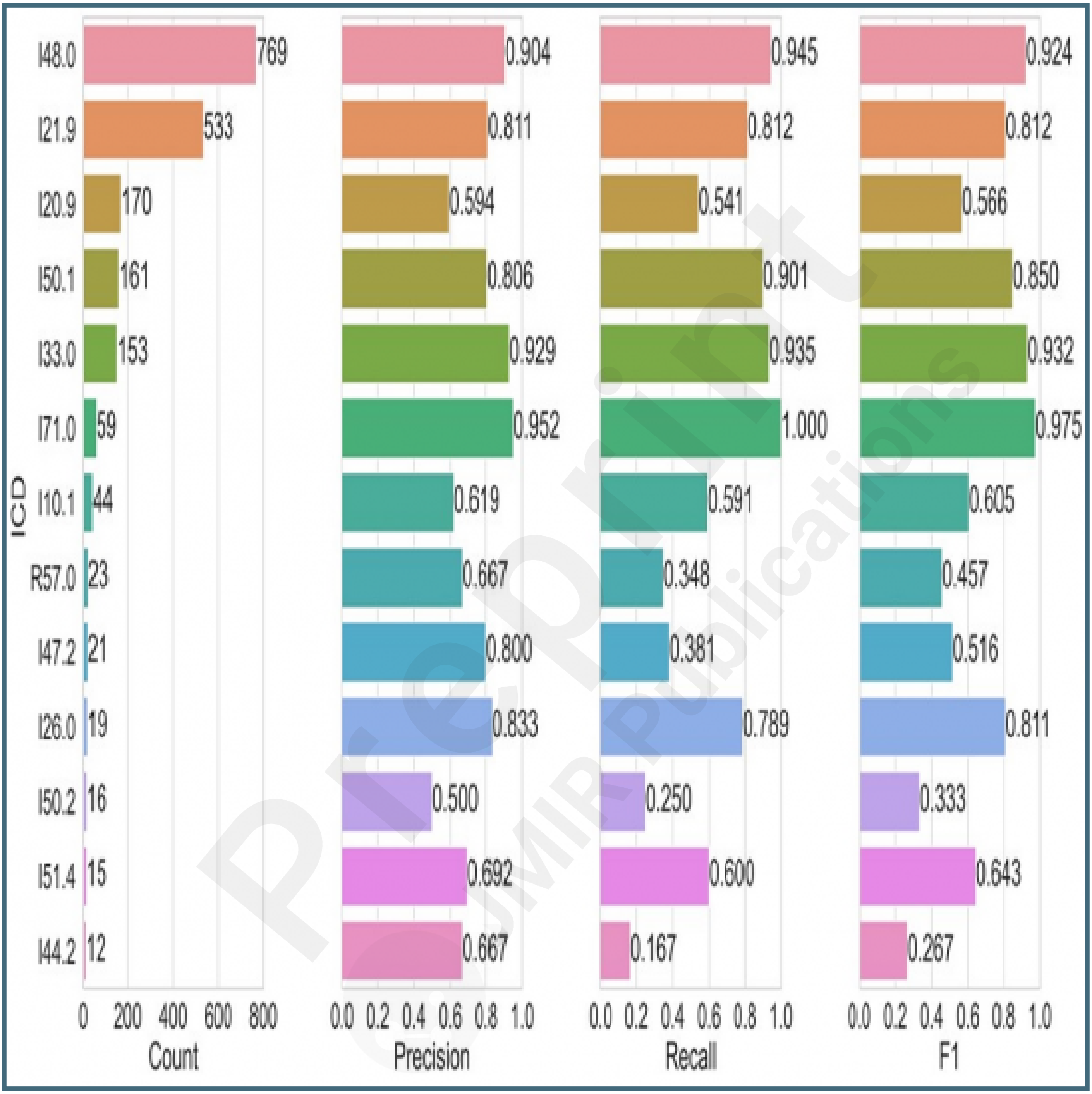
The optimal hyperparameters and their search space.



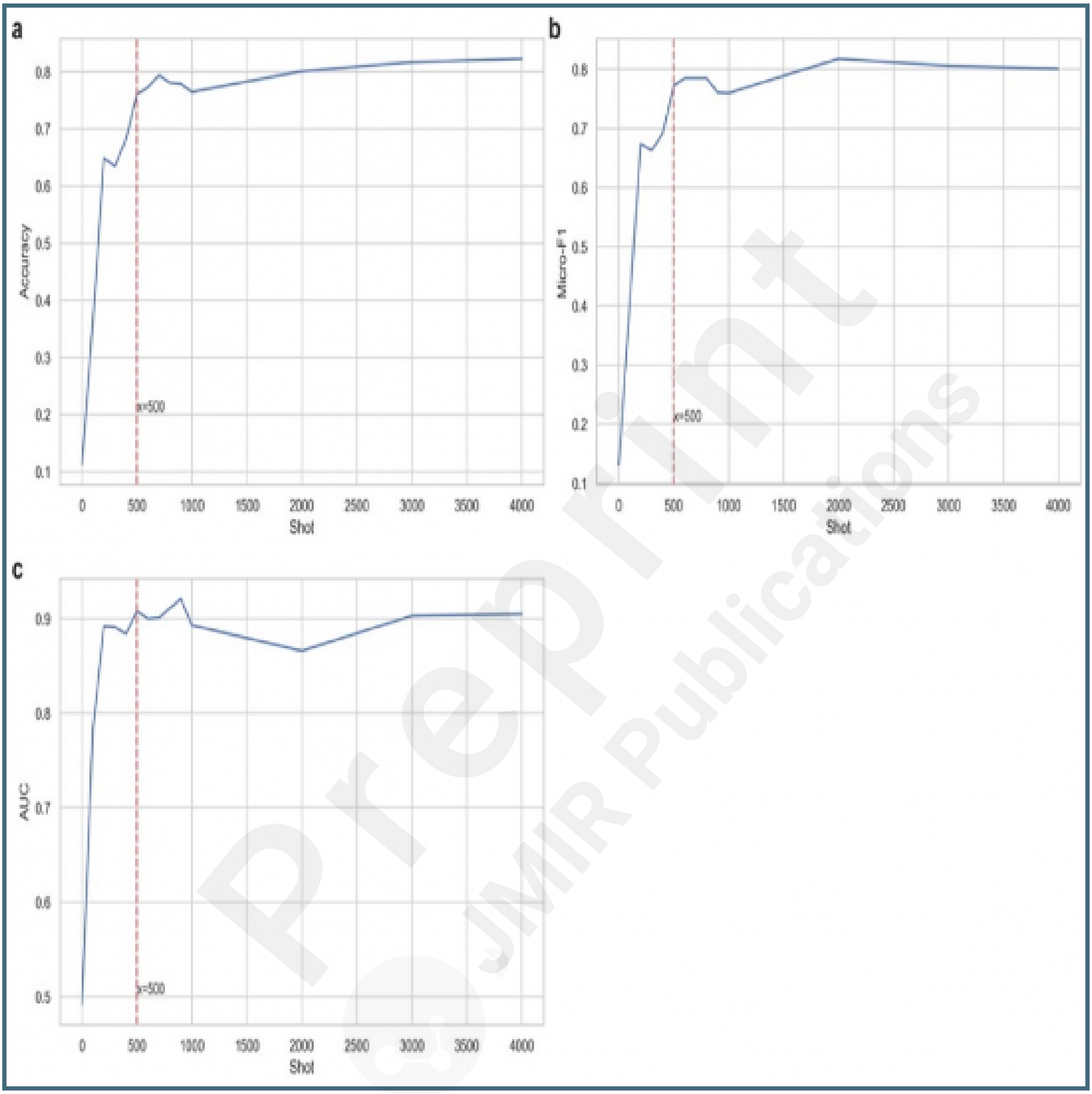
Comparison among different prompt combinations in verbalizer and template.



Precision, recall, and f1 scores of every ICD code in pipeline of MDR-BERT with fine-tuning and prompt learning.



Few shots experiments on MDR-BERT with fine-tuning and prompt learning.



Multimedia Appendixes

Overview of target ICD codes and diseases names.
URL: http://asset.jmir.pub/assets/3325b7a390923a0b4ae394703c51be8a.xlsx

The optimal hyperparameters and their search space.
URL: http://asset.jmir.pub/assets/6b54b9c12e9c1ae3b7f9ab858effeef2.xlsx