

PLM-ICD: Automatic ICD Coding with Pretrained Language Models

Chao-Wei Huang^{*†} Shang-Chi Tsai^{*} Yun-Nung Chen^{*}

^{*}National Taiwan University, Taipei, Taiwan

[†]Taiwan AI Labs, Taipei, Taiwan

f07922069@csie.ntu.edu.tw y.v.chen@ieee.org

Abstract

Automatically classifying electronic health records (EHRs) into diagnostic codes has been challenging to the NLP community. State-of-the-art methods treated this problem as a multi-label classification problem and proposed various architectures to model this problem. However, these systems did not leverage the superb performance of pretrained language models, which achieved superb performance on natural language understanding tasks. Prior work has shown that pretrained language models underperformed on this task with the regular fine-tuning scheme. Therefore, this paper aims at analyzing the causes of the underperformance and developing a framework for automatic ICD coding with pretrained language models. We spotted three main issues through the experiments: 1) large label space, 2) long input sequences, and 3) domain mismatch between pretraining and fine-tuning. We propose **PLM-ICD**, a framework that tackles the challenges with various strategies. The experimental results show that our proposed framework can overcome the challenges and achieves state-of-the-art performance in terms of multiple metrics on the benchmark MIMIC data.¹

1 Introduction

The clinical notes in electronic health records (EHRs) are written as free-form text by clinicians during patient visits. The notes can be associated with diagnostic codes from the International Classification of Diseases (ICD), which represent diagnostic and procedural information of the visit. The ICD codes are a standardized way to encode information systematically and internationally, which could be used for tracking healthcare statistics, quality outcomes, and billing.

While ICD codes provide several useful applications, manually labelling ICD codes has been

shown to be very labor-intensive and domain expertise is required (O'malley et al., 2005). Hence, automatically assigning ICD codes to clinical notes has been of broad interest in the medical natural language processing (NLP) community. Prior work has identified several challenges of this task, including the large number of labels to be classified, the long input sequence, and the imbalanced label distribution, i.e., the long-tail problem (Xie et al., 2019). These challenges make the task extremely difficult, demonstrating that advanced modeling techniques are required. With the introduction of deep learning models, we have seen tremendous performance improvement on the task of automatic ICD coding (Shi et al., 2017; Xie and Xing, 2018; Mullenbach et al., 2018; Li and Yu, 2020; Vu et al., 2020; Cao et al., 2020; Liu et al., 2021; Kim and Ganapathi, 2021; Zhou et al., 2021). These methods utilized convolutional neural networks (CNNs) (Mullenbach et al., 2018; Li and Yu, 2020; Liu et al., 2021) or recurrent neural networks (RNNs) (Vu et al., 2020) to transform the long text in clinical notes into hidden representations. State-of-the-art methods employed a label attention mechanism, i.e., performing attention to hidden representations independently for each label, to combat the extremely large label set (Mullenbach et al., 2018; Vu et al., 2020).

Recently, pretrained language models (PLMs) with the Transformer (Vaswani et al., 2017) architecture have become the dominant forces for NLP research, achieving superior performance on numerous natural language understanding tasks (Devlin et al., 2019; Liu et al., 2019). These models are pretrained on large amount of text with various language modeling objectives, and then fine-tuned on the desired downstream tasks to perform different functionalities such as classification (Devlin et al., 2019) or text generation (Radford et al., 2019; Raffel et al., 2020).

While PLMs demonstrate impressive capabili-

¹The source code is available at <https://github.com/MiuLab/PLM-ICD>.

ties across classification tasks, applying PLMs to automatic ICD coding is still not well-studied. Previous work has shown that applying PLMs to this task is not straightforward (Zhang et al., 2020; Pascual et al., 2021), and the main challenges being:

- The length of clinical notes exceeds the maximum length of PLMs.
- The regular fine-tuning scheme where we add a linear layer on top of the PLMs does not perform well for multi-label classification problems with a large label set.
- PLMs are usually pretrained on general-domain corpora, while clinical notes are very medical-specific and the language usage is different.

As a result, the performance of PLMs reported in the prior work is inferior to the state-of-the-art models that did not use pre-trained models by a large margin (Pascual et al., 2021). Their best model achieved 88.65% in terms of micro-AUC, compared with the state-of-the-art 94.9% from the ISD model (Zhou et al., 2021). This result highlighted that the performance of PLMs on this task was still far from the conventional models.

In this paper, we aim at identifying the challenges met during applying PLMs to automatic ICD coding and developing a framework that could overcome these challenges. We first conduct preliminary experiments to verify and investigate the challenges mentioned above, and then we propose proper mechanisms to tackle each challenge. The proposed mechanisms are: 1) domain-specific pre-training for the domain mismatch problem, 2) segment pooling for the long input sequence problem, and 3) label attention for the large label set problem. By integrating these techniques together, we propose **PLM-ICD**, a framework specifically designed for automatic ICD coding with PLMs. The effectiveness of PLM-ICD is verified through experiments on the benchmark MIMIC-3 and MIMIC-2 datasets (Saeed et al., 2011; Johnson et al., 2016). To the best of our knowledge, PLM-ICD is the first Transformer-based pretrained language model that achieves competitive performance on the MIMIC datasets. We further analyze several factors that affect the performance of PLMs, including pretraining method, pretraining corpora, vocabulary construction, and optimization schedules.

The contributions of this paper are 3-fold:

- We perform experiments to verify and analyze the challenges of utilizing PLMs on the task of automatic ICD coding.
- We develop **PLM-ICD**, a framework to fine-tune PLMs for ICD coding, that achieves competitive performance on the benchmark MIMIC-3 dataset.
- We analyze the factors that affect PLMs’ performance on this task.

2 Related Work

2.1 Automatic ICD Coding

ICD code prediction is a challenging task in the medical domain. Several recent work attempted to approach this task with neural models. Choi et al. (2016) and Baumel et al. (2018) used recurrent neural networks (RNN) to encode the EHR data for predicting diagnostic results. Li and Yu (2020) recently utilized a multi-filter convolutional layer and a residual layer to improve the performance of ICD prediction. On the other hand, several work tried to integrate external medical knowledge into this task. In order to leverage the information of definition of each ICD code, RNN and CNN were adopted to encode the diagnostic descriptions of ICD codes for better prediction via attention mechanism (Shi et al., 2017; Mullenbach et al., 2018). Moreover, the prior work tried to consider the hierarchical structure of ICD codes (Xie and Xing, 2018), which proposed a tree-of-sequences LSTM to simultaneously capture the hierarchical relationship among codes and the semantics of each code. Also, Tsai et al. (2019) introduced various ways of leveraging the hierarchical knowledge of ICD by adding refined loss functions. Recently, Cao et al. (2020) proposed to train ICD code embeddings in hyperbolic space to model the hierarchical structure. Additionally, they used graph neural network to capture the code co-occurrences. LAAT (Vu et al., 2020) integrated a bidirectional LSTM with an improved label-aware attention mechanism. EffectiveCAN (Liu et al., 2021) integrated a squeeze-and-excitation network and residual connections along with extracting representations from all encoder layers for label attention. The authors also introduced focal loss to tackle the long-tail prediction problem. ISD (Zhou et al., 2021) employed extraction of shared representations among high-frequency and low-frequency codes and a self-distillation learning mechanism to alleviate the

long-tail code distribution. Kim and Ganapathi (2021) proposed a framework called Read, Attend, and Code (RAC) to effectively predict ICD codes, which is the current state-of-the-art model on this task. Most recent models focused on developing an effective interaction between note representations and code representations (Cao et al., 2020; Zhou et al., 2021; Kim and Ganapathi, 2021). Our work, instead, is focusing on the choice of the note encoder, where we apply PLMs for their superior encoding capabilities.

2.2 Pretrained Language Models

Using pretrained language models to extract contextualized representations has led to consistent improvements across most NLP tasks. Notably, ELMo (Peters et al., 2018) and BERT (Devlin et al., 2019) showed that pretraining is effective for both LSTM and transformer (Vaswani et al., 2017) models. Variants have been proposed such as XLNet (Yang et al., 2019), RoBERTa (Liu et al., 2019). These models are pretrained on large amount of general domain text to grasp the capability to model textual data, and fine-tuned on common classification tasks.

To tackle domain-specific problems, prior work adapted such models to scientific and biomedical domains, including BioBERT (Lee et al., 2019), ClinicalBERT (Alsentzer et al., 2019), PubMedBERT (Gu et al., 2020) and RoBERTa-PM (Lewis et al., 2020). These models are pretrained on domain-specific text carefully crawled and processed for improving the downstream performance. The biomedical-specific PLMs reported improved performance on a variety of biomedical tasks, including text mining, named entity recognition, relation extraction, and question answering (Lee et al., 2019).

While PLMs achieved state-of-the-art performance on various tasks, applying PLMs to large-scale multi-label classification is still a challenging research direction. Chang et al. (2019) proposed X-BERT, a framework that is scalable to an extremely large label set of a million labels. Lehečka et al. (2020) showed that the modeling capacity of BERT’s pooling layers might be limited for automatic ICD coding. Pascual et al. (2021) also demonstrated inferior performance when applying BERT to this task and pointed out several challenges to be addressed. Specifically, the authors proposed 5 truncation and splitting strategies

Model	Length	Macro-F	Micro-F
LAAT	4000	9.9	57.5
	512*	6.8	47.3
BERT	512*	2.8	38.9

Table 1: Results of LAAT and BERT on MIMIC-3 with different maximum input lengths (%). *The length is number of words for LAAT and number of tokens for BERT, so their performance cannot directly comparable.

to tackle the long input sequence problem. Their proposed *All* splitting strategies is similar to our segment pooling mechanism. However, without the label attention mechanism, the model failed to learn.

Zhang et al. (2020) proposed BERT-XML, an extension of BERT for ICD coding. The model was pretrained on a large cohort of EHR clinical notes with an EHR-specific vocabulary. BERT-XML handles long input text by splitting it into chunks and performs prediction for each chunk independently with a label attention mechanism from AttentionXML (You et al., 2019). The predictions are finally combined with max-pooling. Our proposed framework, PLM-ICD, shares a similar idea with BERT-XML that we also split clinical notes into segments to compute segment representations. The main difference is that we leverage an improved label attention mechanism and we use document-level label-specific representations rather than chunk level representations as in BERT-XML. In Section 5, we demonstrate that PLM-ICD can achieve superior results on the commonly used MIMIC-3 dataset compared with BERT-XML.

3 Challenges for PLMs

In this section, we discuss 3 main challenges for PLMs to work on automatic ICD coding and conduct experiments to verify the severity of the challenges.

3.1 Long Input Text

Pretrained language models usually set a maximum sequence length as the size of their positional encodings. A typical value is set to 512 tokens after subword tokenization (Devlin et al., 2019). However, clinical notes are long documents which often exceed the maximum length of PLMs. For instance, the average number of words in the MIMIC-

Model	Codes	Macro-F	Micro-F
LAAT	50	66.6	71.5
	Full	9.9	57.5
BERT	50	61.5	65.4
	Full	3.2	40.9

Table 2: Results of LAAT and BERT on MIMIC-3 with full codes and top-50 codes (%).

3 dataset is 1,500 words, or 2000 tokens after sub-word tokenization.

To demonstrate that this is a detrimental problem for PLMs, we conduct experiments on MIMIC-3 where the input text is truncated to 512 words for the strong model LAAT (Vu et al., 2020), and 512 tokens for BERT. The results are shown in Table 1. Both models perform worse when the input text is truncated, showing that simple truncation does not work for the long input text problem. Note that the same trend can be found for other models for ICD coding. The results reported by Pascual et al. (2021) also show similar problem where the truncation methods such as *Front-512* and *Back-512* performed much worse than models with longer input context.

3.2 Large Label Set

Automatic ICD coding is a large-scale multi-label text classification (LMTC) problem, i.e., finding the relevant labels of a document from a large set of labels. There are about 17,000 codes in ICD-9-CM and 140,000 codes in ICD-10-CM/PCS, while there are 8921 codes presented in the MIMIC-3 dataset. PLMs utilize a special token and extract the hidden representation of this token to perform classification tasks. For example, BERT uses a [CLS] token and adds a pooling layer to transform its hidden representation into a distribution of labels (Devlin et al., 2019). However, while this approach achieves impressive performance on typical multi-class classification tasks, it is not very suitable for LMTC tasks. Lehečka et al. (2020) showed that making predictions based on only the representation of [CLS] token results in inferior performance compared with pooling representations of all tokens, and hypothesized that this is due to the lack of modeling capacity of using the [CLS] token alone.

To examine the PLMs’ capability of performing LMTC, we conduct experiments on MIMIC-3

in two settings, Full and Top-50. The Full setting uses the full set of 8,921 labels, while the Top-50 uses the top-50 most frequent labels. We report the numbers for LAAT directly from Vu et al. (2020). For the BERT model, we use the segment pooling mechanism to handle the long input, which is detailed in Section 4.2. We aggregate the hidden representations of the [CLS] token for each segment with mean-pooling as the document representation. The final prediction is obtained by transforming the document representation with a linear layer.

The results are shown in Table 2. BERT achieves slightly worse performance than LAAT in the Top-50 setting. However, in the Full setting, BERT performs significantly worse compared with LAAT. The results suggest that using BERT’s [CLS] token for LMTC is not ideal, and advanced techniques for LMTC are required for PLMs to work on this task.

3.3 Domain Mismatch

Normally, PLMs are pretrained on large amount of general-domain corpora which contains billions of tokens. The corpora is typically crawled from Wikipedia, novels (Zhu et al., 2015), webpages, and web forums. Prior work has shown that the domain mismatch between the pretraining corpus and the fine-tuning tasks could degrade the downstream performance (Gururangan et al., 2020).

Specifically for the biomedical domain, several pretrained models have been proposed which are pretrained on biomedical corpora to mitigate the domain mismatch problem (Lee et al., 2019; Alsentzer et al., 2019; Gu et al., 2020; Lewis et al., 2020). These models demonstrate improved performance over BERT on various medical and clinical tasks, showing that domain-specific pretraining is essential to achieve good performance.

4 Proposed Framework

The task of ICD code prediction is formulated as a multi-label classification problem (Kavuluru et al., 2015; Mullenbach et al., 2018). Given a clinical note of $|d|$ tokens $\mathbf{d} = \{t_1, t_2, \dots, t_{|d|}\}$ in EHR, the goal is to predict a set of ICD codes $\mathbf{y} \subseteq \mathcal{Y}$, where \mathcal{Y} denotes the set of all possible codes. Typically, the labels are represented as a binary vector $\mathbf{y} \in \{0, 1\}^{|\mathcal{Y}|}$, where each bit y_i indicates whether the corresponding label is presented in the note.

The proposed framework **PLM-ICD** is illus-

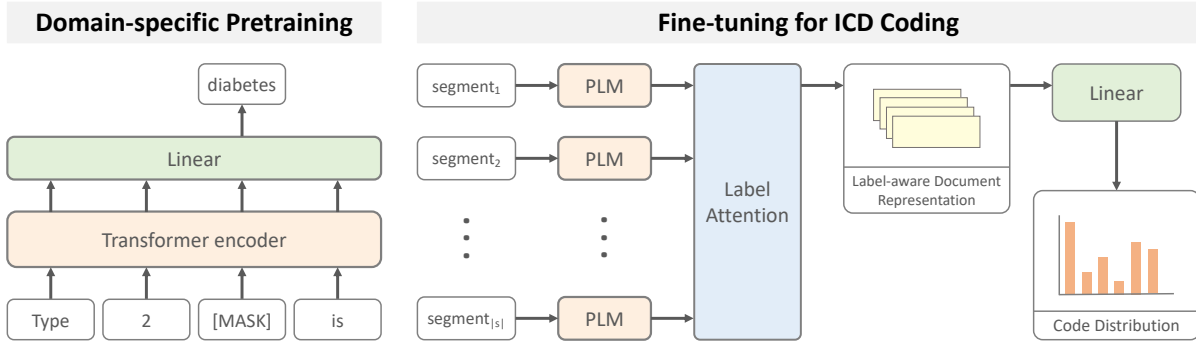


Figure 1: Illustration of our proposed framework. Left: domain-specific pretraining, where a PLM is pretrained on text from specific domains with a language modeling objective. Right: PLM encodes segments of a document separately, and a label-aware attention mechanism is to aggregate the segment representations into label-aware document representations. The document representations are linear-transformed to predict ICD codes.

trated in Figure 1. The details of the components are described in this section.

4.1 Domain-Specific Pretraining

Automatic ICD coding is a domain-specific task where the input text consists of clinical notes written by clinicians. The clinical notes contain many biomedical terms, and understanding these terms is essential in order to assign ICD codes accurately. While general PLMs are pretrained on large amount of text, the pretraining corpora usually does not contain biomedical text, not to mention clinical records.

In order to mitigate the domain mismatch problem, we propose to utilize the PLMs that are pretrained on biomedical and clinical text, e.g., BioBERT (Lee et al., 2019), PubMedBERT (Gu et al., 2020), and RoBERTa-PM (Lewis et al., 2020). These PLMs are specifically pretrained for biomedical tasks and proven to be effective on various downstream tasks. We take the domain-specific PLMs and fine-tune them on the task of automatic ICD coding. We can plug-and-play the domain-specific PLMs since their architectural design and pretraining objective are identical to their general-domain counterparts. This makes our framework agnostic to the type of PLMs, i.e., we can apply any transformer-based PLMs as the encoder.

4.2 Segment Pooling

In order to tackle the long input text problem described in Section 3.1, we propose **segment pooling** to surpass the maximum length limitation of PLMs. The segment pooling mechanism first splits the whole document into segments that are shorter

than the maximum length, and encodes them into segment representations with PLMs. After encoding segments, the segment representations are aggregated as the representations for the full document.

More formally, given a document $d = \{t_1, t_2, \dots, t_{|d|}\}$ of $|d|$ tokens, we split it into $|s|$ consecutive segments s_i of c tokens:

$$s_i = \{t_j \mid c \cdot i \leq j < c \cdot (i + 1)\}$$

The segments are fed into PLMs separately to compute hidden representations, then concatenated to obtain the hidden representations of all tokens:

$$\mathbf{H} = \text{concat}(PLM(s_1), \dots, PLM(s_{|s|}))$$

The token-wise hidden representations \mathbf{H} can then be used to make prediction based on the whole document.

4.3 Label-Aware Attention

To combat the problem of a large label set, we propose to augment the PLMs with the label-aware attention mechanism proposed by Vu et al. (2020) to learn label-specific representations that capture the important text fragments relevant to certain labels. After the token-wise hidden representations \mathbf{H} are obtained, the goal is to transform \mathbf{H} into label-specific representations with attention mechanism.

The label-aware attention takes \mathbf{H} as input and compute $|\mathcal{Y}|$ label-specific representations. This mechanism can be formulated into 2 steps. First, a label-wise attention weight matrix \mathbf{A} is computed

as:

$$\begin{aligned}\mathbf{Z} &= \tanh(\mathbf{V}\mathbf{H}) \\ \mathbf{A} &= \text{softmax}(\mathbf{W}\mathbf{Z})\end{aligned}$$

where \mathbf{V} and \mathbf{W} are linear transforms. The i^{th} row of \mathbf{A} represents the weights of the i^{th} label, and the softmax function is performed for each label to form a distribution over all tokens. Then, the matrix \mathbf{A} is used to perform a weighted-sum of \mathbf{H} to compute the label-specific document representation:

$$\mathbf{D} = \mathbf{H}\mathbf{A}^\top$$

where \mathbf{D}_i represents the document representations for the i^{th} label.

Finally, we use the label-specific document representation \mathbf{D} to make predictions:

$$\mathbf{p}_i = \text{sigmoid}(\langle \mathbf{L}_i, \mathbf{D}_i \rangle)$$

where \mathbf{L}_i is a vector for the i^{th} label, $\langle \cdot \rangle$ represents inner product between two vectors, \mathbf{p}_i is the predicted probability of the i^{th} label. Note that the inner product could also be seen as a linear transform with output size 1. We can then assign labels to a document based on a predefined threshold t .

The training objective is to minimize the binary cross-entropy loss $\mathcal{L}(\mathbf{y}, \mathbf{p})$:

$$-\frac{1}{|\mathbf{y}|} \sum_{i=1}^{|\mathbf{y}|} \left(\mathbf{y}_i \log \mathbf{p}_i + (1 - \mathbf{y}_i) \log(1 - \mathbf{p}_i) \right).$$

5 Experiments

In order to evaluate the effectiveness of our proposed framework, we conduct experiments and compare the results with the prior work.

5.1 Setup

We evaluate PLM-ICD on two benchmark datasets for ICD code prediction.

- **MIMIC-2** To be able to directly compare with the prior work (Mullenbach et al., 2018; Li and Yu, 2020; Vu et al., 2020), we evaluate PLM-ICD on the MIMIC-2 dataset (Saeed et al., 2011). We follow the setting from Mullenbach et al. (2018), where 20,533 summaries are used for training, and 2,282 summaries are used for testing. There are 5,031 labels in the dataset.

- **MIMIC-3** The Medical Information Mart for Intensive Care III (MIMIC-3) (Johnson et al., 2016) dataset is a benchmark dataset which contains text and structured records from a hospital ICU. We use the same setting as Mullenbach et al. (2018), where 47,724 discharge summaries are used for training, with 1,632 summaries and 3,372 summaries for validation and testing, respectively. There are 8,922 labels in the dataset.

The preprocessing is done by following the steps described in Mullenbach et al. (2018) with their provided scripts². Detailed training setting is provided in Appendix A.

5.2 Evaluation

We evaluate our methods with commonly used metrics to be directly comparable to previous work. The metrics used are macro F1, micro F1, macro AUC, micro AUC, and precision@K, where $K = \{5, 8, 15\}$.

5.3 Results

We present the evaluation results in this section. All the reported scores are averaged over 3 runs with different random seeds. The results of the compared methods are taken directly from their original paper. We mainly compare our model, PLM-ICD, with the models without special code description modeling. The performance of models with special code description modeling, i.e., HyperCore, ISD, and RAC, are also reported for reference.

5.3.1 MIMIC-3

The results on MIMIC-3 full test set are shown in Table 3. PLM-ICD achieves state-of-the-art performance among all models in terms of micro F1 and all precision@k measures, even though we do not leverage any code description modeling. All the improvements are statistically significant. RAC performs best on AUC scores and macro F1. We note that the techniques proposed by RAC are complementary to our framework, and it is possible to add the techniques to further improve our results. However, this is out of the scope of this paper.

5.3.2 MIMIC-2

The results on MIMIC-2 test set are shown in Table 4. PLM-ICD achieves state-of-the-art performance among all models in terms of micro F1 and

²<https://github.com/jamesmullenbach/caml-mimic>

Model	AUC		F1		P@k		
	Macro	Micro	Macro	Micro	P@5	P@8	P@15
CAML (2018)	89.5	98.6	8.8	53.9	-	70.9	56.1
DR-CAML (2018)	89.7	98.5	8.6	52.9	-	69.0	54.8
MultiResCNN (2020)	91.0	98.6	8.5	55.2	-	73.4	58.4
LAAT (2020)	91.9	98.8	9.9	57.5	81.3	73.8	59.1
JointLAAT (2020)	92.1	98.8	10.7	57.5	80.6	73.5	59.0
EffectiveCAN (2021)	91.5	98.8	10.6	58.9	-	75.8	60.6
PLM-ICD (Ours)	92.6 _(.2)	98.9 _(.1)	10.4 _(.1)	59.8 [†] _(.3)	84.4 [†] _(.2)	77.1 [†] _(.2)	61.3 [†] _(.1)
<i>Models with Special Code Description Modeling</i>							
HyperCore (2020)	93.0	98.9	9.0	55.1	-	72.2	57.9
ISD (2021)	93.8	99.0	11.9	55.9	-	74.5	-
RAC (2021)	<i>94.8</i>	<i>99.2</i>	<i>12.7</i>	<i>58.6</i>	82.9	75.4	60.1

Table 3: Results on the MIMIC-3 full test set (%). The best scores among models without special code description modeling are marked in **bold**. The best scores among all models are *italic*. The values in the parentheses are the standard variation of runs. † indicates the significant improvement with $p < 0.05$.

Model	AUC		F1		P@k		
	Macro	Micro	Macro	Micro	P@5	P@8	P@15
CAML (2018)	82.0	96.6	4.8	44.2	-	52.3	-
DR-CAML (2018)	82.6	96.6	4.9	45.7	-	51.5	-
MultiResCNN (2020)	85.0	96.8	5.2	46.4	-	54.4	-
LAAT (2020)	86.8	97.3	5.9	48.6	64.9	55.0	39.7
JointLAAT (2020)	87.1	97.2	6.8	49.1	65.2	55.1	39.6
PLM-ICD (Ours)	86.8 _(.2)	97.3 _(.1)	6.1 _(.1)	50.4 [†] _(.2)	67.3 [†] _(.2)	56.1 [†] _(.2)	39.9 _(.2)
<i>Models with Special Code Description Modeling</i>							
HyperCore (2020)	88.5	97.1	7.0	47.7	-	53.7	-
ISD (2021)	<i>90.1</i>	<i>97.7</i>	<i>10.1</i>	49.8	-	<i>56.4</i>	-

Table 4: Results on the MIMIC-2 test set (%). EffectiveCAN (2021) and RAC (2021) did not report results on MIMIC-2. The best scores among models without special code description modeling are marked in **bold**. The best scores among all models are *italicized*. The values in the parentheses are the standard variation of the runs. † indicates that the improvement is statistically significant with $p < 0.05$.

all precision@k measures, similar to the results on MIMIC-3. All the improvements are statistically significant except for P@15.

In sum, these results show that PLM-ICD is generalizable to multiple datasets, achieving state-of-the-art performance on multiple metrics on both MIMIC-3 and MIMIC-2.

6 Analysis

This section provides analysis on factors that affect PLM’s performance on automatic ICD coding.

Model	Macro-F	Micro-F
PLM-ICD	10.4	59.8
(a) - domain pretraining	8.9	54.2
(b) - segment pooling	7.2	54.6
(c) - label attention	4.6	48.0

Table 5: Ablation results on the MIMIC-3 full test set (%).

6.1 Ablation Study

To verify the effectiveness of the proposed techniques, we conduct an ablation study on MIMIC-3 full test set. The results are presented in Table 5.

The first ablation we perform is discarding

Model	Macro-F	Micro-F	\hat{F}
RoBERTa-PM	10.4	59.8	1.35
BioBERT	9.1	57.9	1.60
ClinicalBERT	8.8	57.8	1.60
PubMedBERT	9.2	59.5	1.41

Table 6: Results with different PLMs on the MIMIC-3 full test set (%). \hat{F} is the fragmentation ratio.

domain-specific pretraining. In this setting, we use the pretrained `RoBERTa-base` model as the PLM, and fine-tune it for ICD coding. As shown in row (a), the performance slightly degrades after discarding domain-specific pretraining. This result demonstrates that domain-specific pretraining contributes to the performance improvement.

The second ablation we perform is discarding segment pooling. In this setting, we replace our segment pooling with the one proposed by [Zhang et al. \(2020\)](#). They applied label attention and made code predictions for each segment separately, and aggregated the predictions with max-pooling. As shown in row (b), replacing our segment pooling results in worse performance. This result indicates that our proposed segment pooling is more effective for aggregating segment representations.

The third ablation is removing the label attention mechanism. We fall back to the normal PLM paradigm, i.e., extracting representations of the `[CLS]` token for classification. This setting is identical to the one described in Section 3.2, where we aggregate the representation of the `[CLS]` token for each segment with mean-pooling, and obtain the final prediction by transforming the aggregated representation with a linear layer. As shown in row (c), removing label attention mechanism results in huge performance degradation. The micro F1 score degrades by 11.8% absolute, while the macro F1 score degrades more than half. This result demonstrates that the label attention mechanism is crucial to ICD coding, which is an observation aligned with the prior work ([Mullenbach et al., 2018](#)).

6.2 Effect of Pretrained Models

While we have shown that domain-specific pretraining is beneficial to ICD coding, we would like to explore which domain-specific PLM performs the best on this task. We conduct experiments with different PLMs, including BioBERT ([Lee et al., 2019](#)), ClinicalBERT ([Alsentzer et al., 2019](#)), PubMedBERT ([Gu et al., 2020](#)), and RoBERTa-PM ([Lewis](#)

Model	Macro-F	Micro-F
LAAT	10.4	59.8
CAML	8.7	58.1
BERT-XML	8.2	56.9

Table 7: Results with different attention mechanisms on the MIMIC-3 full test set (%).

Model	Macro-F	Micro-F
Ours	10.4	59.8
HIER-BERT	2.8	42.7
Longformer	5.1	51.6

Table 8: Results with different strategies for tackling the long input problem on the MIMIC-3 full test set (%).

[et al., 2020](#)).

The results are presented in Table 6. RoBERTa-PM achieves the best performance among the 4 examined PLMs. This result is in line with the reported results on the BLURB leaderboard ([Gu et al., 2020](#)), which is a collection of biomedical tasks.

We also report the fragmentation ratio, i.e., the number of tokens per word after subword tokenization as ([Chalkidis et al., 2020](#)). We observe that the PLMs with vocabulary trained on biomedical texts (RoBERTa-PM and PubMedBERT) perform better than the ones inherited vocabulary from BERT-base (BioBERT and ClinicalBERT). The fragmentation ratio also shows that models with custom vocabulary suffer less on the over-fragmentation problem.

6.3 Effect of Label Attention Mechanisms

We conduct experiments with different label attention mechanisms and report the results in Table 7. We compare the label attention mechanisms from LAAT ([Vu et al., 2020](#)), CAML ([Mullenbach et al., 2018](#)) and BERT-XML ([Zhang et al., 2020](#)). The results show that the label attention used in LAAT is best-suited to our framework.

6.4 Effect of Long Input Strategies

We also conduct experiments to verify the effect of different strategies for tackling the long input problem. As shown in Table 8, our proposed segment pooling outperforms HIER-BERT ([Chalkidis et al., 2019](#)) and Longformer ([Beltagy et al., 2020](#)), demonstrating the effectiveness of our proposed method.

Max Length	Segment Length	Macro-F	Micro-F
6144	128	9.2	60.0
3072	256	9.4	59.2
3072	128	9.2	59.6
3072	64	8.2	59.3
3072	32	6.9	57.8

Table 9: Results with different maximum lengths on the MIMIC-3 full dev set (%).

6.5 Effect of Maximum Length

We conduct experiments where we alter the maximum length of the documents and segments to explore the different choices of maximum lengths. The results are shown in Table 9.

When fixing the maximum length of the documents to 3,072, we observe that longer segments results in better performance until the segment length reaches 128. Using a longer maximum document length such as 6144 results in slightly better performance. However, longer sequences require more computation. Considering the trade-off between computation and accuracy, we set maximum document length to 3,072 and segment length to 128 as our defaults.

6.6 Effect of Optimization Process

Similar to the prior work (Sun et al., 2019), we also notice that the fine-tuning process is sensitive to the hyperparameters of the optimization process, e.g., batch size, learning rate, and warmup schedule.

With several preliminary experiments conducted on these factors, we observe that the learning rate and the warmup schedule greatly affects the performance. When we reduce learning rate to $2e-5$, the model performs 3% worse than using the default parameters in terms of micro F1. The warmup schedule is crucial in our framework. When we use constant learning rate throughout training, the model performs about 4% worse. We do not observe clear difference between different scheduling strategies.

6.7 Best Practices

With the above analyses, we provide a guideline and possible future directions for applying PLMs to ICD coding or tasks with similar properties:

- With the input length exceeding the maximum length of PLMs, segment pooling can be used to extract representations of all tokens. PLMs

with longer input length or recurrence could be explored in the future.

- The representation of the [CLS] token might be insufficient when dealing with LMTC problems. A label attention mechanism could be beneficial in such scenarios.
- The pretraining corpora plays an important role for domain-specific tasks.
- The hyperparameters of the optimization process greatly affect the final performance, so trying different parameters is preferred when the performance is not ideal.

7 Conclusion

In this paper, we identify the main challenges of applying PLMs on automatic ICD coding, including the long text input, the large label set and the mismatched domain. We propose **PLM-ICD**, a framework with PLMs that tackles the challenges with various techniques. The proposed framework achieves state-of-the-art or competitive performance on the MIMIC-3 and MIMIC-2 datasets. We then further analyze factors that affect PLMs’ performance. We hope this work could open up the research direction of leveraging the great potential of PLMs on ICD coding.

Acknowledgements

We thank reviewers for their insightful comments. This work was financially supported from the Young Scholar Fellowship Program by Ministry of Science and Technology (MOST) in Taiwan, under Grants 111-2628-E-002-016 and 111-2634-F-002-014.

References

- Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, and Matthew McDermott. 2019. [Publicly available clinical BERT embeddings](#). In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 72–78, Minneapolis, Minnesota, USA. Association for Computational Linguistics.
- Tal Baumel, Jumana Nassour-Kassis, Raphael Cohen, Michael Elhadad, and Noemie Elhadad. 2018. Multi-label classification of patient notes: case study on icd code assignment. In *Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence*.
- Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv:2004.05150*.

- Pengfei Cao, Yubo Chen, Kang Liu, Jun Zhao, Shengping Liu, and Weifeng Chong. 2020. [HyperCore: Hyperbolic and co-graph representation for automatic ICD coding](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3105–3114, Online. Association for Computational Linguistics.
- Ilias Chalkidis, Ion Androutsopoulos, and Nikolaos Aletras. 2019. [Neural legal judgment prediction in English](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4317–4323, Florence, Italy. Association for Computational Linguistics.
- Ilias Chalkidis, Manos Fergadiotis, Sotiris Kotitsas, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. [An empirical study on large-scale multi-label text classification including few and zero-shot labels](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7503–7515, Online. Association for Computational Linguistics.
- Wei-Cheng Chang, Hsiang-Fu Yu, Kai Zhong, Yiming Yang, and Inderjit Dhillon. 2019. Taming pretrained transformers for extreme multi-label text classification. *arXiv preprint arXiv:1905.02331*.
- Edward Choi, Mohammad Taha Bahadori, Andy Schuetz, Walter F Stewart, and Jimeng Sun. 2016. Doctor AI: Predicting clinical events via recurrent neural networks. In *Machine learning for healthcare conference*, pages 301–318. PMLR.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *NAACL*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2020. Domain-specific language model pretraining for biomedical natural language processing. *arXiv preprint arXiv:2007.15779*.
- Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. [Don't stop pretraining: Adapt language models to domains and tasks](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.
- Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. MIMIC-III, a freely accessible critical care database. *Scientific data*, 3:160035.
- Ramakanth Kavuluru, Anthony Rios, and Yuan Lu. 2015. An empirical evaluation of supervised learning approaches in assigning diagnosis codes to electronic medical records. *Artificial intelligence in medicine*, 65(2):155–166.
- Byung-Hak Kim and Varun Ganapathi. 2021. Read, attend, and code: Pushing the limits of medical codes prediction from clinical notes by machines. In *Machine Learning for Healthcare Conference*, pages 196–208. PMLR.
- Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. [BioBERT: a pre-trained biomedical language representation model for biomedical text mining](#). *Bioinformatics*.
- Jan Lehečka, Jan Švec, Pavel Ircing, and Luboš Šmídl. 2020. Adjusting bert's pooling layer for large-scale multi-label text classification. In *International Conference on Text, Speech, and Dialogue*, pages 214–221. Springer.
- Patrick Lewis, Myle Ott, Jingfei Du, and Veselin Stoyanov. 2020. [Pretrained language models for biomedical and clinical tasks: Understanding and extending the state-of-the-art](#). In *Proceedings of the 3rd Clinical Natural Language Processing Workshop*, pages 146–157, Online. Association for Computational Linguistics.
- Fei Li and Hong Yu. 2020. Icd coding from clinical text using multi-filter residual convolutional neural network. In *AAAI*.
- Yang Liu, Hua Cheng, Russell Klopfer, Matthew R Gormley, and Thomas Schaaf. 2021. Effective convolutional attention network for multi-label clinical document classification. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5941–5953.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- James Mullenbach, Sarah Wiegrefe, Jon Duke, Jimeng Sun, and Jacob Eisenstein. 2018. Explainable prediction of medical codes from clinical text. In *NAACL*, pages 1101–1111, New Orleans, Louisiana. Association for Computational Linguistics.
- Kimberly J O'malley, Karon F Cook, Matt D Price, Kimberly Raiford Wildes, John F Hurdle, and Carol M Ashton. 2005. Measuring diagnoses: ICD code accuracy. *Health services research*, 40(5p2):1620–1639.
- Damian Pascual, Sandro Luck, and Roger Wattenhofer. 2021. Towards bert-based automatic icd coding: Limitations and opportunities. *arXiv preprint arXiv:2104.06709*.

- Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. [Deep contextualized word representations](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *Journal of Machine Learning Research*, 21(140):1–67.
- Mohammed Saeed, Mauricio Villarroel, Andrew T Reisner, Gari Clifford, Li-Wei Lehman, George Moody, Thomas Heldt, Tin H Kyaw, Benjamin Moody, and Roger G Mark. 2011. Multiparameter intelligent monitoring in intensive care ii (mimic-ii): a public-access intensive care unit database. *Critical care medicine*, 39(5):952.
- Haoran Shi, Pengtao Xie, Zhiting Hu, Ming Zhang, and Eric P Xing. 2017. Towards automated icd coding using deep learning. *arXiv preprint arXiv:1711.04075*.
- Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune bert for text classification? In *China National Conference on Chinese Computational Linguistics*, pages 194–206. Springer.
- Shang-Chi Tsai, Ting-Yun Chang, and Yun-Nung Chen. 2019. Leveraging hierarchical category knowledge for data-imbalanced multi-label diagnostic text understanding. In *LOUHI*, pages 39–43, Hong Kong. Association for Computational Linguistics.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*, pages 5998–6008.
- Thanh Vu, Dat Quoc Nguyen, and Anthony Nguyen. 2020. A label attention model for icd coding from clinical text. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 3335–3341. International Joint Conferences on Artificial Intelligence Organization.
- Pengtao Xie and Eric Xing. 2018. [A neural architecture for automated ICD coding](#). In *ACL*, pages 1066–1076, Melbourne, Australia. Association for Computational Linguistics.
- Xiancheng Xie, Yun Xiong, Philip S Yu, and Yangyong Zhu. 2019. Ehr coding with multi-scale feature attention and structured knowledge graph propagation. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 649–658.
- Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*.
- Ronghui You, Zihan Zhang, Ziyi Wang, Suyang Dai, Hiroshi Mamitsuka, and Shanfeng Zhu. 2019. Attentionxml: Label tree-based attention-aware deep model for high-performance extreme multi-label text classification. *Advances in Neural Information Processing Systems*, 32:5820–5830.
- Zachariah Zhang, Jingshu Liu, and Narges Razavian. 2020. [BERT-XML: Large scale automated ICD coding using BERT pretraining](#). In *Proceedings of the 3rd Clinical Natural Language Processing Workshop*, pages 24–34, Online. Association for Computational Linguistics.
- Tong Zhou, Pengfei Cao, Yubo Chen, Kang Liu, Jun Zhao, Kun Niu, Weifeng Chong, and Shengping Liu. 2021. Automatic icd coding via interactive shared representation networks with self-distillation mechanism. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5948–5957.
- Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.

A Training Details

We take the pretrained weights released by original authors without any modification. For the best PLM-ICD model, we use RoBERTa-base-PM-M3-Voc released by [Lewis et al. \(2020\)](#). During fine-tuning, we train our models for 20 epochs. AdamW is chosen as the optimizer with a learning rate of $5e - 5$. We employ a linear warmup schedule with 2000 warmup steps, and after that the learning rate decays linearly to 0 throughout training. The batch size is set to 8. All models are trained on a GTX 3070 GPU. We truncate discharge summaries to 3072 tokens due to memory consideration, and the length of each segment c is set to 128. The validation set is used to find the best-performing

threshold t , and we use it to perform evaluation on the test set.