# ClinicalLongformer: Public Available Transformers Language Models for Long Clinical Sequences

**Haoxue Li (hl3664), Xiaocheng Li (xl3119), Chutang Luo (cl5293), Gaomin Wu (gw1107)** *
Center for Data Science
New York University
{hl3664, xl3119, cl5293, gw1107}@nyu.edu

## Abstract

ICD Coding is an important task in medical NLP. Previous state-of-the-arts performances have been reached by neural networks. However, few tries of Transforerms language models (LMs) have been applied to ICD Coding, as a standard Transformers LM only accepts 512 tokens. In this paper we propose to train a Longformer on medical notes, which can accept up to 4096 tokens. Moreover, we compared continue training and initialized Longformer trained on tokenizer built by ourselves, focusing on word-level information of medical notes. We find that Longformer out performs standard BERT-base, and using our initialized tokenizer also improves model's capacity in capturing information required for ICD coding. Eventually, we publish model with the best performance as ClinicalLongformer.

## 1 Introduction

The International Classification of Diseases (ICD) is a healthcare classification system supported by the World Health Organization. It serves as a unique and standardized classification system indicating diseases, symptoms, signs, etc.. ICD codes have been used in a variety of ways, including billing and predicting clinical events [1, 2, 3, 4]. Since manual ICD coding has been demonstrated as expensive, time-consuming, and error-prone, research communities have been studying automatic ICD coding.

Among all methods of automatic ICD coding, methods with deep learning architectures have been the most successful [5, 6, 7, 8, 9]. Currently, ClinicalBERT is the best model for medical notes. However, ClinicalBERT can not deal with long text, as sequence lengths of clinical notes usually exceed the maximum sequence length that a standard Transformers LM accepts. Thus, we need a language model that could take in long text for ICD coding tasks.

In our project, we developed a Longformer model pretrained on medical notes. Meanwhile, considering there are many domain words in medical notes, we also redefine tokenizers and train word embeddings based on medical vocabulary. Our Longformer model with mimic tokenizer outperform BERT-base model on fine-tuning and probing tasks, which proves our model is effective.

## 2 Related Work

**Domain Specific Transformers LMs** It is common to apply Transformers LMs in specific domain. [10] pretrained FinBERT using a large scale of financial communication corpora. SCI BERT [11] is pretrained on large multi-domain corpus of scientific publications to improve performance on downstream scientific NLP tasks. BioBERT [12] is pretrained on large-scale biomedical corpora. In medical fields, ClinialBERT models are pretrained on clinical notes [13, 14]. However, most of previous work focus on pretraining models on domain data, and none of them has redefined

*Equal contribution. In alphabetic order.

domain tokenizers and word-embedding. More specifically, currently there is no such well-designed Transformers LMs for long-text mimic medical notes.

Table 1: Results on MIMIC-III, 50 labels from previous work.[7].

| |
|---|
| *998.32: Disruption of external operation wound* |
| ... wound infection, and **wound breakdown** ... |
| *428.0: Congestive heart failure* |
| ... DIAGNOSES: 1. **Acute congestive heart failure** |
| 2. Diabetes mellitus 3. Pulmonary edema ... |
| *202.8: Other malignant lymphomas* |
| ... a 55 year-old female with **non Hodgkin's lymphoma** |
| and acquired C1 esterase inhibitor deficiency ... |
| *770.6: Transitory tachypnea of newborn* |
| ... Chest x-ray was consistent with **transient tachypnea** |
| **of the newborn** ... |
| *424.1: Aortic valve disorders* |
| ... mild **aortic stenosis with an aortic valve area** of |
| 1.9 cm squared and 2+ **aortic insufficiency** ... |

**Automatic ICD Coding.** Automatic ICD coding is challenging and important in the medical informatics community, and has been studied extensively with traditional machine learning methods [5, 15] and neural network methods [16, 17]. Specific to ICD coding with discharge summary, a hierarchical SVM is used [5]. More recently, a method using a CNN followed by an attention layer becomee the common baseline architecture for ICD coding [6], with a variety of methods based on it [7, 8]. In terms of senteces encoders except for CNN, Bi-LSTM is proposed [9], followed by label-wise attention mechanism, and a hierachical classifier that both predicts the chapter of an ICD code and the code itself.

**Pretraning Transformers LMs for Long Text** To tackle the problem that normal Transformers LMs are unalbe to process long sequence, the model Longformer is released [18]. More methods are proposed to reduce computational compexity on Transformers LMs, but does not mention the situation of long texts[19]. In this paper we pretrain two Longformers, one continue trained on medical notes, and one initialized with our tokenizer on medical vocabulary.

**Long Text Classification with Transformers LMs.** Standard Transformers LMs accepts up to 512 tokens of the sequence, which fails to meet lengths of long documents with more than 512 tokens. To tackle the problem of long-text classification with standard transformers, one method is to separate texts into non-overlapping snippets, pass them respectively to Transformers LM, and aggregate the $[\mathrm{CLS}]$ representations by a recurrent layer or a transformer layer before feeding the final sentence-level representation to a fully-connected layer [20]. In terms of discharge summaries, A similar method is aggregating $[\mathrm{CLS}]$ representations [21]. Specific to medical notes, one way is to utilize rule-based method to extract useful snippets before passing each of them into a Transformers LM, and aggregate the snippet-level representations through attention layers [22].

## 3 Problem Definition and Algorithm

### 3.1 Task

Automatic ICD Coding is our task we aim to tackle. We approach Automatic ICD Coding as a multi-label classification problem. Given a discharge summary $x$, we learn a function to map $x$ to set of labels $y = [y_1, y_2, ..., y_M]$ where $y_j \in [0, 1]$ and $M$ is the number of ICD Coding classes.

### 3.2 Algorithm

Our approach is to first train a pretrained model in MIMIC notes and then fine-tune the pretrained model for Automatic ICD Coding. The general pipeline is shown in figure 1

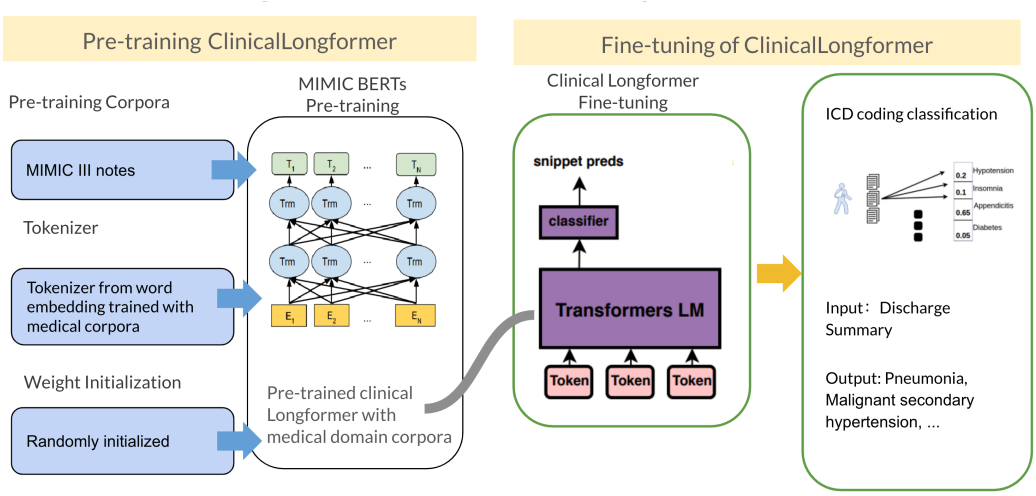



Figure 1: Pretraining and Fine-tuning Pipeline

### 3.2.1 Models Pretraining

In order to tackle the problem that normal Transformers are unable to process long sequence, we use Longformer [18] to represent input text. In order to learn a higher quality text representation in clinical notes, we define our own vocabulary in medical field and tokenzier. We also trained new word embeddings based on the new vocabulary. Our proposed pretrained model: MIMIC BERTs is trained from scratch with Longformer and the new word embeddings. As comparison, we also train BERT from scratch with the new word embeddings and Longformer with default tokenizer. In this section, we will introduce each part of models pretraining in details. We name our pretrained Longformer with our tokenizer as ClinicalLongformer.

**Longfomers** Longformer is a modified Transformer architecture. The original Transformer model has a self-attention component with $\mathcal{O}(n^2)$ time and memory complexity where $n$ is the input sequence length. Longformer uses an attention mechanism that scales linearly with sequence length, making it easy to process documents of thousands of tokens or longer. Longformer's attention mechanism combines a local windowed attention with a task motivated global attention. Longformer is pretrained with masked language modeling (MLM), where tokens from the input sequence are randomly replaced with a [MASK] token and the model learns to recover the randomly masked tokens in a sequence. Note that though Longformer only looks at attention in a fixed-size window, high-level representations can also be captured when multiple Transformers encoder layers are put together.

**Tokenizer Defined on Medical Vocabulary** Longformer is pretrained on $text8$ and $enwik8$ using RoBERTa's WordPiece tokenizer. Despite the fact that the pretrained Longformer consistently outperforms RoBERTa on long document tasks, we train Longformer models from scratch on MIMIC III Clinical Notes. The reason is that clinical notes data contains a specific vocabulary that is not common within a general pretraining corpus like $text8$ and $enwik8$, which would lead to many out of vocabulary(OOV) words. Most pretrained transformers including BERT, RoBERTa and Longformer handles this problem with WordPiece tokenization where OOV words are chunked into sub-words contained in the vocabulary. This may cause the pretrained models in clinical notes learn only to complete the chunked word rather than understand the wider context which decrease the quality of text representation [23, 24]. Hence the first step in our model pretraining process is to re-define tokenizer for the vocabulary in medical field. We call our tokenizer MIMIC tokenizer.

**Word Embeddings trained with fastText Model** As we use MIMIC tokenizer defined on medical vocabulary, we need to train word embeddings based on the new vocabulary. The new word embeddings is used in the Transformers LMs. We use the skipgram model provided by fastText [25] to learn word embeddings.

**BERT** As comparison, we also train BERT from scratch with the new word embeddings. BERT [26] is an encoder composed of stacked transformer[27] modules, which consists of self-attention, normalization, and position-wise fully connected layers. BERT is pretrained with both a masked language model task as well as a next sentence prediction task, where in the latter, the model produce a binary classification to predict if one sentence follows another, in order to capture model longer term dependencies.

### 3.2.2 Fine-tuning Models for ICD Coding Classification

In models pretraining process, we train three models: i) Our proposed pretrained model: MIMIC BERTs which is trained from scratch with Longformer and the new tokenizer. ii) Longformer with default tokenizer. iii) BERT-based model which is trained from scratch with BERT and the new tokenizer. Then, we fine-tune each model for downstream task to evaluate training effects. The downstream task is to disease terms (ICD codes) classification.

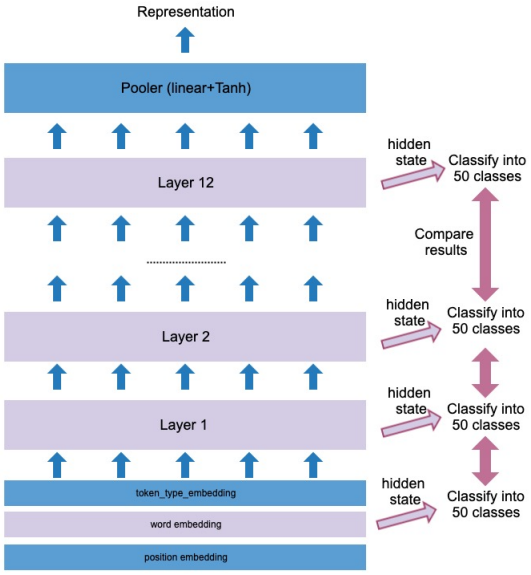



Figure 2: Layer-wise probing architecture.

### 3.2.3 Layer-wise Probing

In order to better understand the pretrained model and to quantify where information regarding the target task is captured within the pretrained model, we include a layer-wise probing task on BERT-base and Longformers we have pretrained. We also set ICD coding as our target task. Specifically, we measure task performance at different layers based on the probing method similar to Edge Probing [28]. However, we do not mix the representations of different layers, but only considers the last layer of interest. We are interested in evaluating how well information of a target task can be extracted from layers a pre-trained encoder.

After the model $f_\theta$ is pretrained trained, we extract representations generated after each layers with respect to input sequences, i.e. $\{h_0, h_1, ..., h_{12}\}$, $h_i \in \mathcal{R}^{n \times d}$ where $n$ is the sequence length and $d$ is the hidden dimension size. For example $h_0$ denotes word embeddings and $h_{12}$ denotes the output representations of the last layer. Then we use embeddings extracted from each Transformer layer (layers 0 up to 12) from the pretrained MIMIC BERTs as the inputs of the probing classifiers $c$. The probing classifiers of each layer are trained to generate predictions $p_i$ based on representations yielded in different layers, i.e. $p_i = c(h_i)$. For this paper we simply consider a affine transformation followed by a Sigmoid function as our classifier architecture. See figure 2 for the architecture.

We use differences in layer-wise probing to quantify contributions of each layer to this classification task and to reveal which layers contain most information for a particular task.

# 4 Experiments setup

## 4.1 Data

In this paper, we use the third version of **M**edical **I**nformation **M**art for **I**ntensive **C**are (MIMIC-III) [29].

### 4.1.1 MIMIC-III

We use all the categories of MIMIC-III to pretrain and fine-tune our models. MIMIC-III is an openly available dataset developed by the MIT Lab for Computational Physiology, comprising de-identified health data associated with 60,000 intensive care unit admissions. It includes demographics, vital signs, laboratory tests, medications, and more. MIMIC-III contains of 15 categories of data, of which we use all data for pretraining, but only consider discharge summaries for our fine-tuning task. Details of texts lengths in different categories are shown in figure (TODO).

### 4.1.2 Preprocessing

We first replace masks for anonymity issues with "unknown name, location, ..." , filter all the brackets and replace the abbreviation. Then, we use sci-spacy package to tokenize sentences. After separating each punctuation from words, we replace numbers with [num] token and replace words that are not in our vocabulary with [unk] token. Finally, we consider each training/validation sample to be a single paragraph inside the sentence.

Following previous work, we use discharge summaries for fine-tuning version, which condensed all information of a stay into a single document. We have some admissions have multiple stays in discharge summary in MIMIC-III, for which we concatenate them following previous work [6]. There are 8,921 ICD codes in total. For this paper we only consider top-50 codes by frequency.

**Train Test Splitting** In pretraining we split the data based on patients. The fraction of training set and validation set is 0.98:0.02. In fine-tuning we follow previous work [7] to split data.

## 4.2 Tokenizer and Word Embeddings

We train our word embeddings with FastText. In choosing vocabulary we consider only words and punctuations splitted by spaces. We select the top 50,000 tokens by frequency, considering that the rest of all tokens have low frequencies, and are likely to be typos. We add special tokens required for Transformers LMs, and get a vocabulary of 50,004 tokens.

## 4.3 Hyperparameters

For pretraining, we did some hyperparameter search in the beginning. Finally we decide to perform pretraining with batch size 32, using Adam optimizer with learning rate 1e-5 and epsilon 2e-8. We train BERT-base with MIMIC tokenizer for 32,000 steps, ClinicalLongformer for 16,000 steps, and Longformer with default RoBERTa tokenizer for 14,000 steps. Note that we will continue pretraining our models in the future, as overfitting is hard to define in pretraining Transformers LMs.

For fine-tuning we use batch size 32 and Adam optimizer with learning rate 2e-5 and epsilon 2e-8.

For layer-wise probing, we use the same batch size 32 and Adam optimizer, but with different learning rate 0.01 and epsilon 2e-8. Larger learning rate results in more efficient converge speed in validation loss.

## 4.4 Evaluation Metrics

In pretraining, we use perplexity (PPl) and bits per character (BPC) as our evaluation metrics. However, perplexity shows little evidence of how well do pretrained models work since we train models on different tokenizers. Therefore, we follow evaluation metrics in [29] to test model performances on fine-tuning. We also include differences in layer-wise probing to test contributions of each layer to this classification task.

## 5 Results and Discussion

### 5.1 Word Embedding

We visualize the word embedding we get, which is shown in figure 3

Figure 3: Visualization of Word Embedding

Our word embedding really makes sense, because words that have similar context are close to each other, which is in line with the principles of FastText model. We further examine our results by evaluating the nearest and farthest words of a given word. We use cosine similarity to calculate the distance of two words in the vocabulary of word embedding.

Table 2: The Top 5 nearest and farthest words of "diabetes"

| Nearest Words | Distance | Farthest Words | Distance |
|---|---|---|---|
| mellitus | 0.8428 | slightly | 0.0087 |
| dm | 0.5889 | changed | 0.0115 |
| hypercholesterolemia | 0.5142 | minimal | 0.0125 |
| dyslipidemia | 0.5054 | air | 0.0146 |
| hyperlipidemia | 0.5009 | arin | 0.0182 |

From table 2, we can see that the nearest words to diabetes are mellitus, dm, hypercholesterolemia, dyslipidemia and hyperlipidemia. Diabetes and mellitus are often written together, and dm is their abbreviation, which is the reason why these two words are the top 2 nearest words to diabetes. hypercholesterolemia, dyslipidemia and hyperlipidemia are symptoms related to diabetes. The farthest words of diabetes has nothing to do with diabetes. This example also proves that our word embedding works.

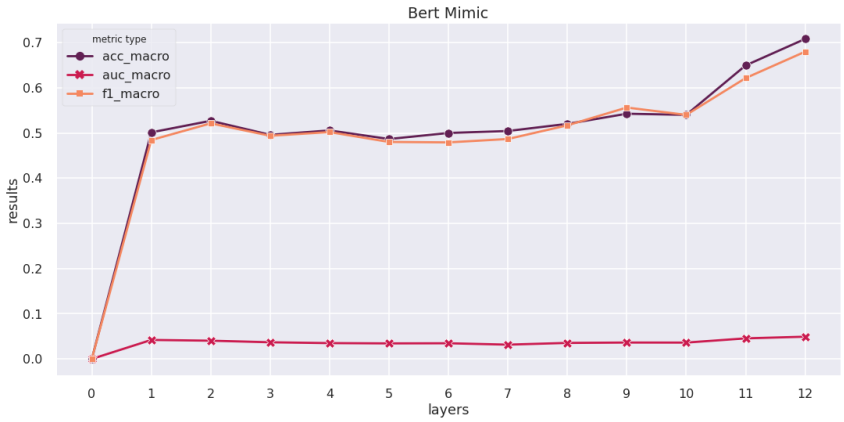



Figure 4: Probing results of pretrained initialized BERT-base with MIMIC tokenizer. Note that layer 0 referes to word embeddings.

### 5.1.1 Pretraining and Fine-tuning models

For each model we report the performance of its fine-tuned model in ICD coding, which includes macro AUC, micro AUC, macro f1 and micro f1, and the performance of the pretrianed model itself: ppl and bpc. The results are shown in Table 3.

Firstly, we find that both Longformer based models outperform BERT based models. We believe this speaks to the benefit of using Longformer based models in EHR notes data. The result shows that models that accepts longer inputs are better in practice, because EHR notes tend to be very long. Secondly, we find that models with MIMIC tokenizer outperform models with default byte-pair encoding (BPE) tokenizer for Transformers. We believe this finding justifies that word-level information works better in medical notes, compared with subwords derived using BPE encoding on regular text data. Last but not least, we find that the improvement of changing default tokenizer to MIMIC tokenizer is greater than using Longformer based models to replace BERT based models. This shows that compared with more advanced model structure, a suitable vocabulary/ tokenizer further improves the performance of in-domain task.

We also notice that within baseline models Longformer-base performs worse than BERT-base, which suggest that without further pretraining, Longformer models cannot recognize the medical terms which are segmented into sub-words. Therefore, longer text inputs means more noise of data in fine-tuning.

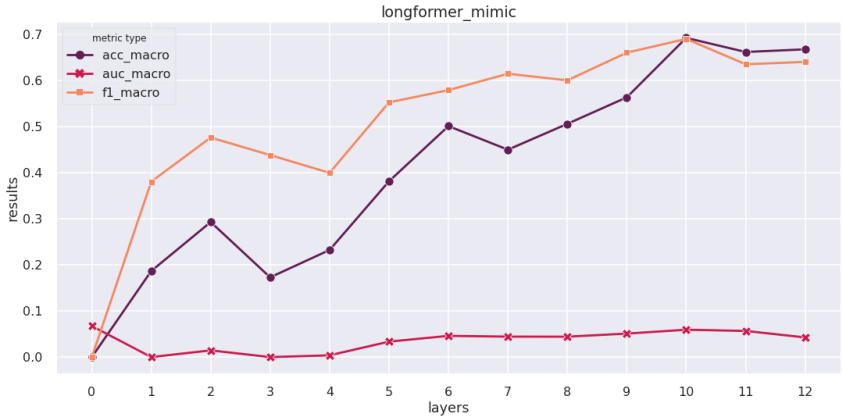
| Model | Tokenizer | # steps | ppl. | bpc | macro auc | micro auc | macro f1 | micro f1 |
|---|---|---|---|---|---|---|---|---|
| BERT-base (baseline) | default | | | | 0.789 | 0.835 | 0.295 | 0.424 |
| Longformer-base (baseline) | default | | | | 0.768 | 0.819 | 0.214 | 0.337 |
| Longformer | default | 16,000 | 4.09 | 2.03 | 0.791 | 0.836 | 0.300 | 0.416 |
| BERT-base | MIMIC | 32,000 | 8.46 | 2.14 | 0.833 | 0.868 | 0.417 | 0.519 |
| **ClinicalLongformer** | **MIMIC** | 14,000 | 22.07 | 4.46 | **0.841** | **0.879** | **0.458** | **0.551** |

Table 3: Pretraining and fine-tuning results. Note that macro auc, micro auc, macro f1 and micro f1 are results on ICD coding.

### 5.1.2 Layer-wise Probing

Figures 4 and 5 show the layer-wise probing results with ICD coding task. As we observe from the probing results, we saw there are little differences across different layers in terms of AUC scores. There are several possible explanations for the low AUC scores, including i) ICD coding task is difficult, so during fine-tuning models need to change embeddings to yield better classification results, and ii) the mean-pooling method to aggregate representations of each position is not ideal for extracting the representation of the sequence. ii) also explains why during layer-wise probing Longformer with default tokenizer outperforms Clinical Longformer. It is also possible that the


(a)




(b)

Figure 5: Probing results of pretrained Longformers (a) continued pretraining on MIMIC-III. (b) initialized with MIMIC tokenizer. Note that layer 0 referes to word embeddings.

probing method we propose is not ideal for the difficult ICD coding task. However, as probing model learning representations of a given task is a research topic far from fully-developed, we think our probing method is acceptable at current stage.

However, when we consider accuracy and F1 scores, we find that models shows a general improvement of probing results. We then show the difference of probing results across layers in figure 6. We consider macro and micro F1 scores, and computes the difference between results of interested layers and corresponding last layers. The results shows that the biggest improvement always occur between the word embedding layer and the first layer, and most improvements occur in the first half of layers (6 layers). It has been mentioned that most tasks which requires shallow linguistic knowledge, including morphological and phrasal knowledge often have probing improvements happening in early layers[28]. Following the argument, we suggest that ICD coding could be a task which only requires shallow linguistic knowledge like word combination. However, having redundant layers does not harm our results in ICD coding, suggesting that some high-level information helps improving ICD coding performance.

We also compare the probing results of different models. We see that the improvement of our pretrained BERT-base with MIMIC tokenizer between layer 0 and layer 1 are often higher than that of Longformer models, with little improvements in F1 scores afterwards. This is because a standard BERT-base can only take 512 tokens as inputs, which largely impacts model to extract useful information that can only be extracted across long sequences. We also find that though our ClinicalLongformer outperforms Longformer with RoBERTa tokenizer in ICD coding task, its performance on probing is worse. One possible explanation for this is the lack of pretraining data and

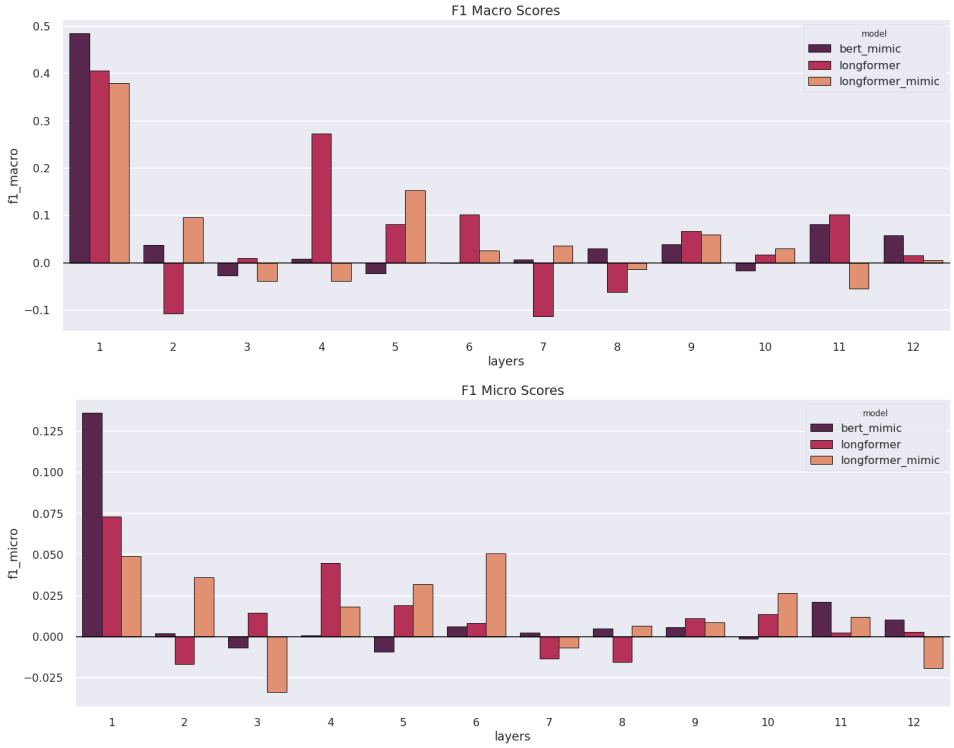



Figure 6: Layers-wise probing results difference with respect to the last layers, considering macro and micro F1 scores. For example, results of layer 1 show the difference between probing in layer 1 and layer 0 (word embeddings). Note that all results have been normalized usint min-max normalization.

pretraining steps. We can expect better performances on probing when we give more pretraining on Longformer.

### 5.2 Future Work

There are some improvements that can be done in the future. First, the data we have used to train domain Transformers LMs is still not enough. It is far less than Wiki Text that have been used to train normal Transformers LMs. We are supposed to collect more medical notes data. Although the computation complexity of LongFormer is proportional to BERT's, it is still too large for long text input. We could try more sparse attention mechanism to improve the architecture of LongFormer. For classification task, it is not very necessary to obtain high-level linguistic knowlege, so a very deep neural network is somehow a waste for this problem. We need to further experiment how many layers we actually need.

In the future, we will continue pretraining models expected to 150k steps. There's little the issue in over-fitting in pretraining LMs, as we can hardly hit the point of overfitting in these large amount of pretraining data. Even though the validation loss reaches a plateau, the model can still learn something if we keep training the models. In addition, we will try to figure out how to extract phrases that contains diseases information for ICD coding. This task is not related to pretraining but only to fine-tuning.

## 6 Conclusion

In this project, our task is to pretrain Transformers LMs used on ICD coding. More generally, we expect this model can also be used in other medical notes problems, but we are mainly interested in classifying medical terms (ICD coding). We find that medical notes are difficult to deal with due to text length and special tokens.

Therefore, we carefully preprocess the input text. We then use medical vocabulary from input text to build a new tokenizer and word embeddings. We pretrain LongFormer, and compare the performance between LongFormer and BERT. We compare the performance of different models on ICD coding tasks. We also perform layer-wise probing on the data.

The word embedding we have trained can effectively reflect the context in medical notes. When fine-tuning pretrained models and baseline models we find that ClinicalLongformer outperforms all other models In probing, we find that in early layers more information realted to ICD coding is encoded, while final layers also contribute to the target task.

## 7 Lessons learned

Traditional Transformers LMs can not do well in medical notes as we have mentioned before. We re-preprocess the data carefully and train LongFormer models to effectively improve performance on downstream tasks. Though we have learned Transformer models like BERT and we know that it can be used for transfer learning, we are not clear about the whole process. From this project, we have known that continue training in domain knowledge is beneficial. Given different tasks and different input text for downstream task, we need to adjust tokenizers and models to get better performance. More specifically, we may need to train new tokenizers and models.

We have gained hands-on experience in building model pipeline, including preprocessing data, developing new tokenizers, training word embeddings, pretraining models and fine-tuning models for downstream tasks, which is quite complex in deep learning domain. We know how to integrate resources to finish our tasks.

Since our dataset is large and the model we have used is relatively more complex, we need more computation resources. To effectively train our model, we have learned how to schedule multiple GPU training high performance computing machine.

Our method proves to be effective for pretraining, fine-tuning and probing as we have discussed above. Though the results of probing is just partially effective, this is mainly due to the drawbacks in current probing methods.

## References


[1] J. Denny, M. Ritchie, M. Basford, J. Pulley, L. Bastarache, K. Brown-Gentry, Deede Wang, D. Masys, D. Roden, and D. Crawford. Phewas: demonstrating the feasibility of a phenome-wide scan to discover gene–disease associations. *Bioinformatics*, 26:1205 – 1210, 2010.

[2] R. Ranganath, A. Perotte, Noémie Elhadad, and D. Blei. The survival filter: Joint survival analysis with a latent time series. In *UAI*, 2015.

[3] Edward Choi, Mohammad Taha Bahadori, Andy Schuetz, Walter F. Stewart, and Jimeng Sun. Doctor ai: Predicting clinical events via recurrent neural networks. In Finale Doshi-Velez, Jim Fackler, David Kale, Byron Wallace, and Jenna Wiens, editors, *Proceedings of the 1st Machine Learning for Healthcare Conference*, volume 56 of *Proceedings of Machine Learning Research*, pages 301–318, Northeastern University, Boston, MA, USA, 18–19 Aug 2016. PMLR.

[4] A. Avati, Kenneth Jung, S. Harman, L. Downing, A. Ng, and N. Shah. Improving palliative care with deep learning. *BMC Medical Informatics and Decision Making*, 18, 2018.

[5] A. Perotte, R. Pivovarov, K. Natarajan, N. Weiskopf, F. Wood, and Noémie Elhadad. Diagnosis code assignment: models and evaluation metrics. *Journal of the American Medical Informatics Association : JAMIA*, 21:231 – 237, 2014.

[6] J. Mullenbach, Sarah Wiegreffe, J. Duke, Jimeng Sun, and Jacob Eisenstein. Explainable prediction of medical codes from clinical text. *ArXiv*, abs/1802.05695, 2018.

[7] F. Li and H. Yu. Icd coding from clinical text using multi-filter residual convolutional neural network. *ArXiv*, abs/1912.00862, 2020.

[8] Pengfei Cao, Yubo Chen, K. Liu, Jun Zhao, Shengping Liu, and Weifeng Chong. Hypercore: Hyperbolic and co-graph representation for automatic icd coding. In *ACL*, 2020.

[9] Thanh Vu, Dat Quoc Nguyen, and A. Nguyen. A label attention model for icd coding from clinical text. In *IJCAI*, 2020.

[10] Yi Yang, Mark Christopher Siy UY, and Allen Huang. Finbert: A pretrained language model for financial communications, 2020.

[11] Iz Beltagy, Kyle Lo, and Arman Cohan. Scibert: A pretrained language model for scientific text. 2019.

[12] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. 2019.

[13] Emily Alsentzer, John R. Murphy, Willie Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, and Matthew B. A. McDermott. Publicly available clinical bert embeddings, 2019.

[14] Kexin Huang, Jaan Altosaar, and Rajesh Ranganath. Clinicalbert: Modeling clinical notes and predicting hospital readmission, 2019.

[15] Ramakanth Kavuluru, Anthony Rios, and Y. Lu. An empirical evaluation of supervised learning approaches in assigning diagnosis codes to electronic medical records. *Artificial intelligence in medicine*, 65 2:155–66, 2015.

[16] Haoran Shi, Pengtao Xie, Zhiting Hu, M. Zhang, and E. Xing. Towards automated icd coding using deep learning. *ArXiv*, abs/1711.04075, 2017.

[17] Pengtao Xie and Eric Xing. A neural architecture for automated ICD coding. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1066–1076, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[18] Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer, 2020.

[19] William Merrill, Vivek Ramanujan, Yoav Goldberg, Roy Schwartz, and Noah Smith. Parameter norm growth during training of transformers, 2020.

[20] Raghavendra Pappagari, Piotr Żelasko, Jesús Villalba, Yishay Carmiel, and Najim Dehak. Hierarchical transformers for long document classification. *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 838–844, 2019.

[21] Andriy Mulyar, Elliot Schumacher, M. Rouhizadeh, and Mark Dredze. Phenotyping of clinical notes with improved document classification models using contextualized neural language models. *ArXiv*, abs/1910.13664, 2019.

[22] Kexin Huang, Sankeerth S. Garapati, and A. Rich. An interpretable end-to-end fine-tuning approach for long clinical text. *ArXiv*, abs/2011.06504, 2020.

[23] Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Ziqing Yang, Shijin Wang, and Guoping Hu. Pre-training with whole word masking for chinese bert. *arXiv preprint arXiv:1906.08101*, 2019.

[24] Zachariah Zhang, Jingshu Liu, and Narges Razavian. Bert-xml: Large scale automated icd coding using bert pretraining. *arXiv preprint arXiv:2006.03685*, 2020.

[25] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

[26] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[27] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30:5998–6008, 2017.

[28] Ian Tenney, Dipanjan Das, and Ellie Pavlick. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy, July 2019. Association for Computational Linguistics.

[29] Alistair E. W. Johnson, Tom J. Pollard, Lu Shen, L. Lehman, M. Feng, M. Ghassemi, Benjamin Moody, Peter Szolovits, L. Celi, and R. Mark. Mimic-iii, a freely accessible critical care database. *Scientific Data*, 3, 2016.

## 8 Contribution

Haoxue Li: fine-tune and probe pretrained models and baselines.

Xiaocheng Li: propose and release ClinicalLongformer and BERT-base with MIMIC tokenizer.

Chutang Luo: developed mimic tokenizer and word embedding.

Gaomin Wu: propose and relsease BERT-base with default tokenizer.

All: preprocess data; write report