Fine-tuning Transformers for Long Clinical Notes

Haau-Sing Li,[1] **Kyunghyun Cho,**[1,2] **and Narges Razavian**[3,4]
[1]Center for Data Science, [2]Dept. of Computer Science, [3]Dept. of Population Health, [4] Dept. of Radiology
New York University
{xl3119, kyunghyun.cho}@nyu.edu, narges.razavian@nyulangone.org

Abstract

Inferring structured EHR data such as disease mentions (ICD codes) from unstructured clinical notes is currently done via manual chart review, which is time consuming and expensive. Previous work on ICD coding with neural networks utilizes Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), or self-attention models based on Transformers, to generate strong textual representations of clinical sentences or documents. Most existing work in ICD coding task is adapted from architectures developed for general domain NLP, and do not address challenges specific to clinical text. In this paper, we propose an adjustments to transformer architecture, to enable handling of discharge summaries, which can span to over tens of thousands of tokens. Our model is based on learning local representations of tokens via n-gram layers and then training self-attention over these local representations. We compare this model to a standard transformer architecture trained over snippets of text. We find that two types of models generate different predictions, hinting that snippet-based and ngram-based models could extract different levels of information for the target task. We then ensemble model predictions, and our ensemble model reaches the highest AUC scores among all sentence encoders, although not outperforming state of the art DR-CAML model.

1 Introduction

The International Classification of Diseases (ICD) using medical notes is a healthcare classification system supported by the World Health Organization. It serves as a unique and standardized classification system indicating diseases, symptoms, and signs. ICD codes have been used in a variety of ways, including billing and predicting clinical events (Denny et al., 2010; Ranganath et al., 2015; Choi et al., 2016; Avati et al., 2018; Liu et al., 2018; Zhang et al., 2020). Since manual ICD coding has been demonstrated as expensive, time-consuming, and error-prone, research communities have been studying automatic ICD coding.

ICD coding is a multi-label classification problem. The task has been considered difficult. First of all, the dimension of label space is often large, with over 15,000 ICD codes from the ICD-9 taxonomy and over 140,000 codes from the ICD-10 taxonomy (Gr, 1988). Furthermore, labels in data are often strongly imbalanced, which further leverage task difficulty. Last but not least, predicting ICD codes with merely texts typed by human specialists is challenging, as the vocabulary within medical notes is different from that of general domain English, and the text data often includes typos, abbreviations, and long sequences non-related to the task.

Among all methods of automatic ICD coding with medical notes, methods using neural networks(NNs) have been the most successful (Mullenbach et al., 2018; Li and Yu, 2020; Cao et al., 2020; Vu et al., 2020). All these models uses simple word embeddings, CNN or RNN to represent sentences, before developing architectures based on these representations.

Transformers LMs have been the most successful LMs regarding a huge number of Natural Language Understanding (NLU) tasks. A number of prior work have also applied Transformers LMs in medical domain, either pretraining LMs or using pretrained LMs for target tasks (Zhang et al., 2020; Alsentzer et al., 2019; Huang et al., 2019). Most existing work have not modified the architecture to allow these models to handle long text, and have relied on text cropping or a rolling-window combined with standard Transformer modeling to handle long notes.

In order to tackle the problem of sentence lengths, we propose an adjustments to transformer architecture, based on learning local representations of tokens via n-gram layers and then training

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** ...

Table 1: Results on MIMIC-III, 50 labels from Li and Yu (2020).

self-attention over these local representations. We compare this model to a standard transformer architecture trained over snippets of text.

Our experiments are on publicly available MIMIC III discharge summary notes, and we predict 50 most common ICD codes per note, following a common benchmark. Our results show that predictions of the two types of models are different, which hints that different levels representations could have different influences on ICD coding. We then ensemble model predictions, and report our results.

2 Related Work

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 (Perotte et al., 2014; Kavuluru et al., 2015) and neural network methods (Shi et al., 2017; Xie and Xing, 2018). Specific to ICD coding with medical notes, Perotte et al. (2014) propose a hierarchical SVM. More recently, Mullenbach et al. (2018) propose a method using a CNN followed by an attention layer. Li and Yu (2020) extend that single-filter CNN to multiple filters with residual connection. Cao et al. (2020) propose a method that utilizes both the method proposed by Mullenbach et al. (2018) and graphical structure within ICD codes. Vu et al. (2020) propose a method that combines a Bi-LSTM, a label-wise attention mechanism, and a hierachical classifier that both predicts the chapter of an ICD code and the code itself. See table 1 for examples of ICD codes followed by text snippets that indicate the presence of the condition.

Representing Sentences with pretrained Transformers LMs. Pretrained Transformers LMs have been more successful than previous architectures such as CNNs and RNNs in various tasks. Candidates of these pretrained models include BERT, RoBERTa, and ALBERT. A series of related work show that pretraining models on any domain can improve the performance of NLP tasks in that same domain (Alsentzer et al., 2019; Gururangan et al., 2020). In clinical domain, a number of transformer models have been pretrained specifically on bio/medical text data (Huang et al., 2019; Alsentzer et al., 2019; Beltagy et al., 2020; Zhang et al., 2020) . However, due to computational limits of current GPUs, most Transformers LMs, even medical ones, only accepts up to 512 tokens.

Feeding Long Text into Transformers LMs for Classification. One approach to applying Transformers to long text is to pretrain models that accepts longer sequences and simultaneously limit computational complexity. Beltagy et al. (2020) pretrain models that accepts sequences up to 4096 tokens, while maintaining computational complexity as same as standard Transformers LMs by limiting nodes that the attention layer visits. However, models pretrained with this method cannot be applied to medical notes, which often have longer texts (>10,000 tokens).

Another approach is to separate input texts into snippets that models can accept, and aggregate resulting representations of all snippets. Pappagari et al. (2019) separate texts into non-overlapping snippets, and aggregate [CLS] representations of every snippet, before applying a recurrent layer or a Transformers layer to get the global representation for long sequences. Mulyar et al. (2019) propose a similar method in aggregating [CLS] representations, but focuses on medical notes. Huang et al. (2020) utilizes rule-based method to extract useful snippets before passing each of them into a Transformers LM, and aggregating the snippet-level representations through attention layers. Note that all these approaches don't address fine-tuning transformers but only utilizes pretrained hidden representations directly. Since for most target tasks fine-tuned transformers generates hidden representations more effective than transformers before fine-tuning, we think fine-tuning is equally important in ICD coding task.

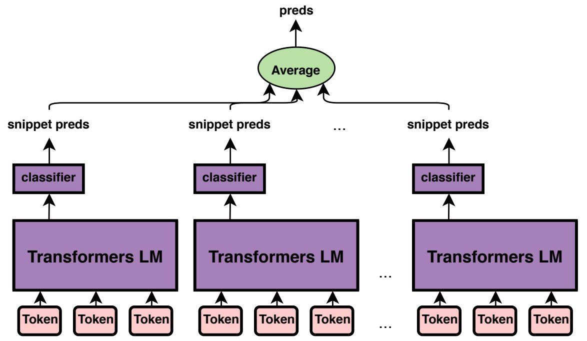

Figure 1: The architecture of snippet-based model. Note that during training, the loss function is computed through snippet-level predictions. We only average predictions for evaluation.

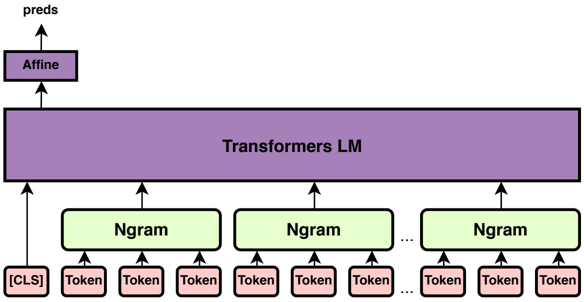

Figure 2: The architecture of ngram-based model.

3 Method

In this section, we introduce our two types of fine-tuned models, i.e. a snippet-based type and an ngram-based type. By such design we assume that they capture dependencies within snippets (local) and across snippets (global) respectively. We then introduce methods of ensembling model predictions. We name our ensemble model **Bi**-directional **En**semble **T**ransf**o**rmers (BEnTo).

3.1 Snippet-Based Model

Let $X = (x_1, x_2, ..., x_T)$ be the long text inputs, and L the dimension of label space. We first extract N snippets of M ($M < 512$) tokens sequentially to cover the whole sequence of discharge summary. Note that we always obtain the last M' tokens from the last snippet.

We pass each snippet to a standard Transformers LM, where we obtain hidden states of snippet i as $H_i^{snippet} = (h_1^i, h_2^i, ..., h_M^i)$. We simply add an affine transformation with a sigmoid function after the [CLS] hidden state h_1^i to obtain predictions $\hat{y_i}^{snippet} \in \mathcal{R}^L$.

During training, we use the corresponding ICD codes of the entire discharge summary sequence as the labels for every snippet extracted from the sequence. In testing time, we simply average the sigmoid of logits of all snippets, i.e.

$$\hat{y}^{snippet} = \frac{1}{N} \sum_{i=1}^{N} \hat{y_i}^{snippet}. \qquad (1)$$

The architecture is shown in figure 1.

3.2 Ngram-based Model

We first pass the entire input data X to get the corresponding pretrained word embeddings $W = (w_1, w_2, ..., w_T)$. After selecting non-overlapping Ngrams, we compute Ngram embeddings by taking the sum of word embeddings within the Ngram of size G. Note that we keep embedding of [CLS], and we only consider non-overlapping Ngrams, i.e. the ith Ngrams's embedding e_i is computed as

$$e_i = \sum_{j=G*(i-1)+2}^{G*(i)+1} w_j. \qquad (2)$$

We then pass the $K + 1$ the Ngram embeddings $E = ([CLS], e_1, ..., e_K)$ to the Transformers LM and get the representations $H^{ngram} = (h_1, h_2, ..., h_{K+1})$. We simply add a fully-connected layer with a sigmoid function after the [CLS] hidden state h_1 to obtain predictions $\hat{y}^{ngram} \in \mathcal{R}^L$. The architecture is shown in figure 2.

3.3 Ensembling Model Predictions

We then have C model predictions (C being the number of snippets in a document), one way of ensembling is to take the element-wise average of the predictions, i.e.

$$\hat{y} = \frac{1}{C} \sum_{c=1}^{C} \hat{y}_c. \qquad (3)$$

During our experiments, we found that our results have low recall, therefore we further modified the ensembling method by taking the element-wise maximum of all predictions, i.e.

$$\hat{y} = \max_{c \in \{1,2,...,C\}} \hat{y}_c. \qquad (4)$$

4 Experiments

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) (Johnson et al., 2016). Note that MIMIC-III is also

	AUC		F1		
Model	Macro	Micro	Macro	Micro	P@5
C-MemNN (Prakash et al., 2017)	0.833	–	–	–	0.42
C-LSTM-Att (Shi et al., 2017)	–	0.900	–	0.532	–
Logistic Regression (Mullenbach et al., 2018)	0.829	0.864	0.477	0.533	0.546
CNN (Mullenbach et al., 2018)	0.876	0.907	**0.576****	0.625	**0.620**
Bi-GRU (Mullenbach et al., 2018)	0.828	0.868	0.484	0.549	0.591
CAML (Mullenbach et al., 2018)	0.875	0.909	0.532	0.614	0.609
DR-CAML (Mullenbach et al., 2018)	0.884	**0.916***	**0.576***	**0.633**	0.618
*snippet-auc	0.874	0.897	0.539	0.578	0.558
*snippet-f1	0.863	0.887	0.527	0.568	0.550
*ngram-auc	0.831	0.871	0.469	0.526	0.521
*ngram-f1	0.818	0.850	0.479	0.539	0.524
snippet-auc-avg	**0.890***	0.910	0.569	0.604	0.576
snippet-f1-avg	0.889	0.907	0.567	0.604	0.579
ngram-auc-avg	0.848	0.886	0.493	0.552	0.543
ngram-f1-avg	0.850	0.884	0.517	0.575	0.558
snippet-all-avg	**0.893***	0.910	**0.570**	**0.608**	0.580
ngram-all-avg	0.857	0.893	0.517	0.573	0.560
all-auc-avg	0.888	0.909	0.544	0.595	0.577
all-f1-avg	0.889	0.910	0.554	**0.606**	0.589
all-all-avg	**0.890**	**0.913**	0.560	**0.608**	0.589

Table 2: Results on MIMIC-III, 50 labels. Our models follow {BEnTo}_{model_type}_{criterion}_{ensembling_method}. The models that comes with the sign * show the average of results of each seed.

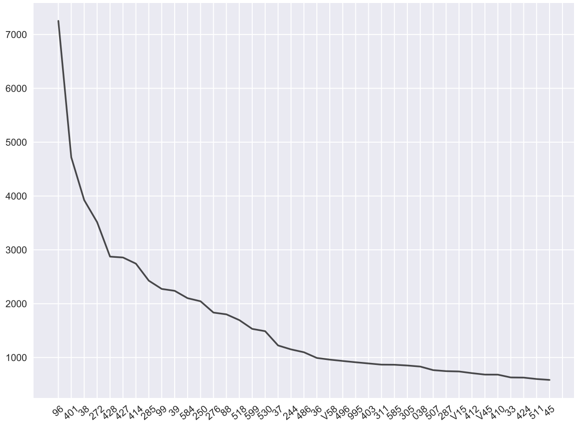

Figure 3: Frequencies of the most frequent 50 ICD Codes.

the pretraining data for ClinicalBERT (Alsentzer et al., 2019).

Medical Notes. Following previous work, we use discharge summaries, which condensed all information of one stay of a patient into a single document. In MIMIC-III, some admissions have multiple stays in discharge summary, for which we concatenate them following Mullenbach et al. (2018). There are 8,921 ICD codes in total. For this paper we only consider top-50 codes ranked by frequency. The frequencies of top-50 ICD codes is shown in figure 3.

Data Preprocessing. We follow Mullenbach et al. (2018) in preprocessing the data. We use ICD-9 codes. We keep discharge summaries that contain at least one of the top-50 codes. As a result, we have 8,067, 1574, 1730 discharge summaries in the training, validation, and test set respectively. By simply splitting tokens with spaces, we get the average text length of 1,797.19, and the maximum text length is 8,417. When we tokenize discharge summaries using standard BERT vocabulary, we get sequences with even more tokens.

4.2 Hyperparameters

Transformers LM. We first test ngram-based models with `bert-base-uncased` and `Bio_Discharge_Summary_BERT`. We find little difference between the performances of two models, then consider only reporting results we experimented with `bert-base-uncased`.

Other hyperparameters. We perform hyperparameter searching of ngram-based models with Ngram size in $\{28, 32, 64, 128\}$. Ngram-based models are trained with 32 samples per batch. For snippet-based models, we set snippet size to be 510, and the number of overlapping tokens 128. Snippet-based models are trained with batch size 16. We train all models with Adam optimizer with learning rate 2e-5 and epsilon 1e-8, and save two checkpoints, one with the highest micro AUC score, and one with highest F1 score.

4.3 Baselines

We compare performance of our models with models that build simple classifier after extracting document-level representations with either CNNs or RNNs. Most results are reported in Mullenbach et al. (2018). However, by either building multiple levels of ICD code representations or inducing external information like the structure of ICD codes, methods proposed by Li and Yu (2020), Cao et al. (2020) and Vu et al. (2020) lead to much better performances in ICD coding. We leave the study of these topics for future work.

C-MemNN The **C**ondensed **Mem**ory **N**eural **N**etwork is proposed by Prakash et al. (2017). They utilizes iterative condensed memory representations in the model.

C-LSTM-Att The **C**haracter-aware **LSTM**-based **Att**ention model is proposed by Shi et al. (2017), which utilizes character-aware LSTMs to generate subsection-wise representations of discharge summaries, and apply attention to match the representations and top-50 codes.

CAML & DR-CAML The **C**onvolutional **A**ttention network for **M**ulti-**L**abel classification was proposed by Mullenbach et al. (2018). CAML utilizes a convolutional sentence encoder and an attention layer before using label-aware representations to perform multi-label classification (McCallum, 1999). The **D**escription **R**egularized CAML is extended CAML with descriptions of ICD codes to regularize model training.

Logistic Regression, CNN, Bi-GRU We use results provided by Mullenbach et al. (2018).

5 Results

In this section, we compared our models with baseline models mentioned above. We first tune Ngram size to find the size that yields the best result. See figure 4 for the learning curve. We find that ngram-based models with smaller Ngram size always perform better. This is quite intuitive as Ngram embeddings preserves more token-level information. We only consider ngram-based model with Ngram size 28 in ensembling. We don't try for shorter Ngrams since with with shorter grams input lengths are likely to surpass the maximum that BERT-base can take.

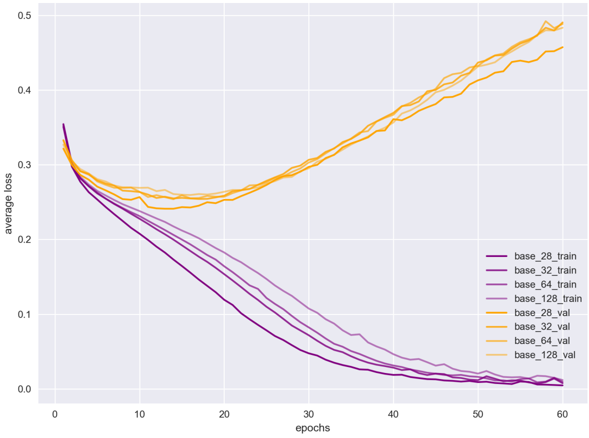

Figure 4: The learning curve of ngram-based model.

We find that ngram-base models overfit at an early stage in terms of cross entropy loss function, but if we continue to train the overfitted models, model predictions decrease in precision and increase in recall, leading to lower AUC scores but higher F1 scores in validation. This happens to both ngram-based and snippet-based model. Therefore, for each model we save one checkpoint with the highest micro AUC score and one with the highest F1 score. We skip plotting the learning curve for snippet-based models as it shows the similar pattern as the learning curves for ngram-based models.

We train each model with 5 random seeds. We report results in table 2, including the average results of all model types and results of all ensemble models. We find that the snippet-based models before ensembling predictions surpasses models with RNN sentence encoders in AUC scores, but fail to match models with CNN encoders. The snippet-based models fails to match models with RNN and CNN encoders in both F1 scores and precion at top 5 ICD codes. One possible explanation is that by only using the snippet-based models, we are unable to utilize information across different text snippets.

We then ensemble model predictions. We only report results with ensembling by taking the average in table 2, and leave the comparison of different

Criterion	Ngram-based	Snippet-based	Cross-model-class
Micro AUC	0.1502	0.1214	**0.1953**
Micro F1	0.2128	0.1395	**0.3020**

Table 3: Within-model-class and across-model-class variations scores for best micro AUC and best micro F1 checkpoints respectively.

	AUC		F1		
Model	Macro	Micro	Macro	Micro	P@5
snippet-auc-avg	0.890	0.910	0.326	0.439	0.576
snippet-auc-max	0.881	0.901	0.440	0.528	0.559
snippet-f1-avg	0.874	0.895	0.481	0.554	0.556
snippet-f1-max	0.889	0.907	0.365	0.467	0.579
ngram-auc-avg	0.848	0.886	0.417	0.510	0.543
ngram-auc-max	0.843	0.881	0.472	0.558	0.535
ngram-f1-avg	0.850	0.884	0.489	0.562	0.558
ngram-f1-max	0.846	0.880	0.499	0.540	0.546
all-f1-avg	0.888	0.910	0.478	0.560	0.589
all-f1-max	0.871	0.897	0.509	0.545	0.552
all-auc-avg	0.884	0.909	0.387	0.497	0.577
all-auc-max	0.870	0.896	0.495	0.563	0.551
all-all-avg	0.890	0.913	0.444	0.532	0.589
all-all-max	0.870	0.897	0.502	0.536	0.552

Table 4: Results with ensembling methods (maximal value, average) on MIMIC-III, 50 labels. All names of model with ensemble predictions follow {model_type}_{criterion}_{ensembling_method}.

ensemble methods in ablation studies. We find that even though ensemble predictions with only ngram-based models have lower AUC scores than CAML, most of the ensemble predictions outperform models with sentences encoders of CNN and RNN, and are comparable to DR-CAML in AUC scores.[1] We also find that ensembling results from ngram-based models and snippet-based models preserve the highest performances whether by AUC, F1 or precision at top-5 codes. This suggests that even by simply averaging the predictions we get from each type of models, we are able to extract different aspects of useful information with respect to the target task. We leave the detailed analysis of generated predictions of different models in Section 6.2.

We note, however, that ensemble predictions have lower F1 scores compared with either CNN, CAML, or DR-CAML since predictions have low recalls. Moreover, in predicting top-5 codes our ensemble results also fail to catch up with results of these methods. One possible explanation is that the training of the snippet-based models could contain a noise as we also feed in non-relevant text snippets with respect to labels during pretraining, and we fails to include the detailed representation of each token in ngram-based models. Another possibile explanation is about the pretrained model which doesn't consider token representations in medical domain, and we leave that for future studies.

6 Model Analysis

6.1 Are Fine-tuned Models Substantially Different?

In the previous section we have shown that ensembling predictions of ngram-based and snippet-based models lead to better performances. We are then interested to see whether predications across models are really different.

We begin with defining variation score within a model class. Given M models within the model class and N samples, the within-model-class variation score is

[1] Note that in the DR-CAML the sentence encoder of CNN is also regularized by the representation of descriptions of ICD codes.

$$\frac{1}{NM(M-1)}\sum_{n=1}^{N}\sum_{m=1}^{M}\sum_{k\neq m}^{M}||f^m(x^n)-f^k(x^n)||_1 \tag{5}$$

where f^m, f^k are models , and $||f^m(x^n) - f^k(x^n)||_1$ is the L1 distance between $f^m(x^n)$ and $f^k(x^n)$.

We also define the cross-model-class variation score. Consider two model class both with M models and N samples, the cross-model-class variation score is

$$\frac{1}{NM^2}\sum_{n=1}^{N}\sum_{m=1}^{M}\sum_{k=1}^{M}||f^m(x^n)-g^k(x^n)||_1. \tag{6}$$

The variation scores are shown in table 3. We notice that:

- checkpoints of best micro AUC have lower variation scores compared with checkpoints of best F1.
- Snippet-based models have lower within-model-class variation scores than ngram-based models.
- Cross-model-class variations scores are higher than within-model-class variation scores.

This suggests that ngram-based models and snippet-based models are generating different predictions. However, whether ngram-based models capture long-term dependencies and snippet-based models capture short-term dependencies remains to be answered.

6.2 Ablation Study: Ensemble Methods

As discussed in Section 5, our ensemble predictions have low F1 scores. One possible explanation could be the wrong choice of method to ensemble predictions. To better understand the effect of ensemble methods, we compare results of ensemble predictions generated by averaging and taking maximal values.

Intuitively, ensembling by taking the maximal predictions sacrifices model precision for recall. As shown in table 4, ensembling by taking maximal predictions leads to higher F1 scores, and lower AUC scores and precision of top-5 codes. Moreover, the increased F1 scores still fails to catch up to CAML, suggesting that the issue of low recall predictions has to do with the model architecture.

7 Discussion and Future Work

Combining Representations within and across Snippets One of our future directions is devising a better mechanism for combining representations within and across text snippets. Our proposed method of averaging the predictions of two models did not fully utilizing the "local" and "global" representations. Huang et al. (2020) suggested using attention mechanism to learn to extract the right level of representation, although they did not fine-tune the transformer layer and used pre-defined rules to extract relevant text snippets. This is a direction for future work.

Classification with Imbalanced Label Distribution As discussed in Section 5 and 6.2, our method generates ICD codes with high precision but low recall. In other words, our method gives accurate prediction results when a label is present, but it is too conservative to tag ICD codes. Moreover, given from precision at 5, for the most frequent ICD codes our method fail to generate precise predictions. This is a challenge due to extreme imbalanced label distribution. Vu et al. (2020) tried to solve this problem by generating label-wise outputs with a hierarchical classification architecture. Another possible solution is to weight labels in the loss function (Kaku et al., 2019). We tried weighting in our pilot study and found that this doesn't work in our target task.

Interpretability As we only add a classifier head on the contextual [CLS] embeddings and consider ensemble predictions of different types of models, generating interpretable results is difficult in our case. Previous works tackles interpretability with different methods (Lei et al., 2016; Li et al., 2016; Ribeiro et al., 2016). Mullenbach et al. (2018) simply picks out the phrase with the highest attention weights given. This method can be applied to our ngram-based model, yet not applicable to the snippet-based. Moreover, as we average predictions for snippet-based models and for ensembling all model predictions, building an explainable classifier becomes a challenge.

Medical Domain-Specific Transformers LMs Medical texts are composed of various types of uncommon tokens compared with pretraining texts for a standard Transformers LM (Devlin et al., 2019; Liu et al., 2019; Lan et al., 2020; Raffel et al., 2020). Gururangan et al. (2020) point out that continuing pretraining on domain-specific text does help a downstream task. We have tried Clinical

BERT(Alsentzer et al., 2019) in our pilot study but can see little difference between models. Since the Clinical BERT is pretrained using the BERT vocabulary, terminologies within the medical domain could be tokenized as combinations of unrecognizable character ngrams. Therefore, another future direction for improvement is to pretrain a domain-specific BERT with the vocabulary within the medical domain.

8 Conclusion

In this paper, we proposed an Ngram based Transformers model, in addition to a standard snippet based model to apply Transformers LMs to ICD coding with long text. We then ensembled model predictions generated by two types of models. We performed experiments on MIMIC-III with top-50 codes, and found that our predictions surpass all other sentence encoders in terms of AUC, but have lower F1 scores. Our future work includes defining a better mechanism for combining representations within and across text snippets.

References

Emily Alsentzer, J. Murphy, Willie Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, and Matthew B. A. McDermott. 2019. Publicly available clinical bert embeddings. *ArXiv*, abs/1904.03323.

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

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *ArXiv*, abs/2004.05150.

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

Edward Choi, Mohammad Taha Bahadori, Andy Schuetz, Walter F. Stewart, and Jimeng Sun. 2016. Doctor ai: Predicting clinical events via recurrent neural networks. In *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. PMLR.

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

J. Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Brämer Gr. 1988. International statistical classification of diseases and related health problems. tenth revision.

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 *ACL*.

Kexin Huang, Jaan Altosaar, and Rajesh Ranganath. 2019. Clinicalbert: Modeling clinical notes and predicting hospital readmission. *arXiv preprint arXiv:1904.05342*.

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

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

Aakash Kaku, Chaitra V. Hegde, Jeffrey Huang, S. Chung, X. Wang, M. Young, Alireza Radmanesh, Y. Lui, and N. Razavian. 2019. Darts: Denseunet-based automatic rapid tool for brain segmentation. *ArXiv*, abs/1911.05567.

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

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. *ArXiv*, abs/1909.11942.

Tao Lei, R. Barzilay, and T. Jaakkola. 2016. Rationalizing neural predictions. In *EMNLP*.

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

J. Li, Xinlei Chen, E. Hovy, and Dan Jurafsky. 2016. Visualizing and understanding neural models in nlp. *ArXiv*, abs/1506.01066.

Jingshu Liu, Zachariah Zhang, and Narges Razavian. 2018. Deep ehr: Chronic disease prediction using medical notes. In *Machine Learning for Healthcare Conference*, pages 440–464.

Y. Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, M. Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

Andrew Kachites McCallum. 1999. Multi-label text classification with a mixture model trained by em. In *AAAI 99 Workshop on Text Learning*.

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

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

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

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

Aaditya Prakash, Siyuan Zhao, Sadid A. Hasan, V. Datla, Kathy Lee, Ashequl Qadir, J. Liu, and Oladimeji Farri. 2017. Condensed memory networks for clinical diagnostic inferencing. *ArXiv*, abs/1612.01848.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, M. Matena, Yanqi Zhou, W. Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

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

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "why should i trust you?": Explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

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

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

Pengtao Xie and Eric Xing. 2018. 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. Association for Computational Linguistics.

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.