Predicting ICD-9 Codes from Medical Notes – Does the Magic of BERT Applies Here?

Stanford CS224N Custom Project (Option 3)
Yiyun Chen
SCPD
Stanford University
starchen@stanford.edu

Abstract

Coding diagnosis and procedures in medical records is tedious and labor intensive, but this process is essential for accurate billings in the current medical system. In this paper, I report the performance of a natural language processing model (BERT) that maps patient discharge notes to ICD codes. Previous studies have demonstrated that deep learning models such as CNN and LSTM perform better at such mapping when compared to conventional machine learning models. Therefore, I employed a transformer-based pretrained model: BERT on the largest emergency department clinical notes dataset MIMIC-III to select for the top-50 ICD codes. The model suffers from the drawback of only allowing the first 510 tokens of a note to be fed into the model so its performance inevitably suffers due to this drawback, especially compare to CNN based methods. Yet somewhat surprisingly, it still outperforms some other baseline models including GRU and ULMFiT which can handle arbitrary length of sequence. For the ICD code prediction, a CNN based model may be a better direction to pursue, but the study still offers some valuable insights on applying BERT-based model to MIMIC-III dataset.

1 Introduction

Assigning International Classification of Diseases (ICD) codes to each patient encounter by medical coders is a tedious process. In the midst of the rise of deep learning, a decent amount of effort have been devoted into creating an automatic mapping between the medical notes and ICD codes. The deep learning methods typically achieve better performance than classical natural language processing (NLP) techniques [1-5]. Most of these studies were conducted before the BERT-era. As one of the recent break-through in NLP, BERT (Bidirectional Encoder Representations from Transformers) is often considered a swiss army knife of NLP that's useful for almost any task due to its pre-training step on a large text corpus [6]. The fine-tuning of BERT is relatively inexpensive and efficient, but its usability hasn't yet been put into test on ICD-code tagging. This study aims to put BERT into inspection regarding its performance on ICD prediction by finetuning a pre-trained BERT model on clinical notes associated with a patients' discharge.

2 Method

ICD-9 code prediction is treated as a multilabel text classification problem. Suppose we have n total number of training examples, and d number of labels. If we let $\mathcal{L}$ be the set of ICD-9 codes, and let $Y \in \{0,1)$ be the set of labels, the prediction is to find a map between each input x_i to its associated class $y_j, where\ i = 1, 2, \ldots, n$ and j = 1, 2, …, d. In a trained classifier, a probability can be associated with each one of the d labels.

2.1 Fine-tuning a pretrained BERT model

A pretrained BERT model was used to predict such a mapping. A BERT model uses a special classification token ([CLS]) concatenated to the beginning of each sequence of training tokens, and use the final hidden state corresponding to this token as the aggregate sequence representation for classification task. The model structure of BERT features a stack of Transformer blocks with self-attention mechanism that allows each token to be associated with any other tokens in the sequence at any point of time. BERT was introduced with two different sizes: $\text{BERT}_{\text{BASE}}$ features 12-layer transformer stacks with 12 attention heads and 110M total parameters; $\text{BERT}_{\text{LARGE}}$ doubles the number of stacks with 16 attention heads and 340M total parameters.

BERT was pretrained for masked language modeling, and for next sentence prediction, where a sentence pair was fed into the model at each time. When using the model for down-stream task like sequence tagging, the input degenerate into a text-Φ pair. The [CLS] representation at the last layer of the transformer stack is fed into an output layer (feed-forward network plus non-linearity) for classification (Figure 1). The output of [CLS] (i.e. $O_{[cls]}$) is a vector of size 768 from $\text{BERT}_{\text{BASE}}$ and size 1024 from $\text{BERT}_{\text{LARGE}}$. It was the only input for the classification layer. Because a patient can be associated with multiple labels, a sigmoid non-linearity was applied to generate independent probability of each label: $\hat{y}_\ell = \sigma(\beta_\ell^T O_{[CLS]} + b_\ell)$. The loss function to minimize is the sum of binary cross-entropy loss over each label: $Loss_{BCE}(X, y) = -\sum_{\ell=1}^{\mathcal{L}}(y_\ell \log(\hat{y}_\ell) + (1 - y_\ell)\log(1 - \hat{y}_\ell))$

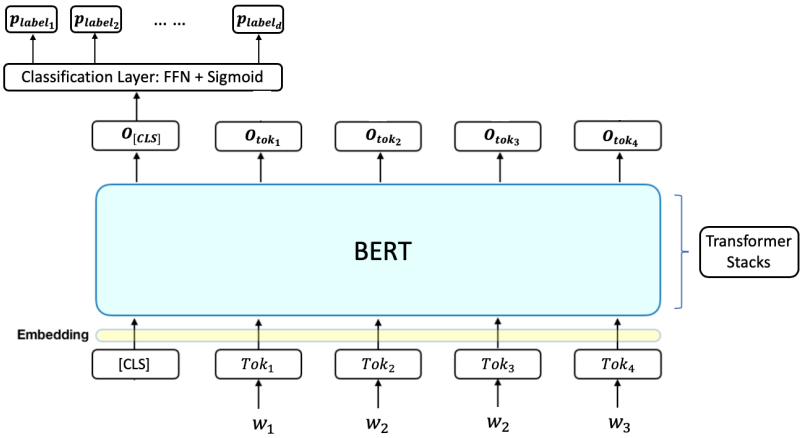

Figure 1 BERT for multi-label sequence classification

There are studies which trained a $\text{BERT}_{\text{BASE}}$ on clinical data (ClinicalBERT, BioBERT, etc.) [REF]. I have compared the performance of using ClinicalBERT/BioBERT versus using the original $\text{BERT}_{\text{BASE}}$ weight as the starting point for finetuning on a subset of sample. The difference is minimal, so I ended up using the original weight of $\text{BERT}_{\text{BASE}}$ to make the results more comparable to $\text{BERT}_{\text{LARGE}}$. The original authors of BERT have suggested a set of hyper-parameter values to choose from. For this study, we picked a learning rate of 2e-5, a batch size of 64 for $\text{BERT}_{\text{BASE}}$ or 16 for $\text{BERT}_{\text{LARGE}}$ on 8 P100 GPUs. The number of batch_size dropped from $\text{BERT}_{\text{BASE}}$ to $\text{BERT}_{\text{LARGE}}$ because the latter requires much larger memory space to store

its 340M parameters as compared to 110M parameters of $BERT_{LARGE}$. The $BERT_{BASE}$ model was trained for 6 epochs and $BERT_{LARGE}$ for 3 epochs.

2.2 Data

Medical Information Mart for Intensive Care (MIMIC) database is an open-access healthcare data of patients admitted to critical care units of a large tertiary care hospital [7]. The database was developed by the Massachusetts Institute of Technology (MIT) containing deidentified Electronic health records (EHR) with both structured and unstructured data including diagnostics and laboratory results, medications, and discharge summaries. Each admission is tagged by human coders with 8,921 unique ICD-9 codes (6,918 diagnosis code and 2,003 procedure codes). I used the same data splitting as in Mullenbach et al. where 47,724 discharge summaries from 36,998 patients for training, 1,632 summaries for validation and 3,372 summaries for testing [4]. Mullenbach tested the model for both the full set of ICD-9 codes, and for the 50 most frequent codes. Given the time limitation, I restricted the scope of study by only focusing on the 50 most frequent codes just to speed things up. And similar to Mullenbach, a smaller testing set was also constructed following Shi et al. (2017) where the test set was filtered down to instances that have at least one of the top 50 most frequent codes, resulting in 8,067 summaries for training, 1,574 for validation, and 1,730 for testing [8]. This is done to make the results comparable across different studies.

2.3 Challenge and Study Re-purpose

A major challenge that appeared during exploratory analysis is that most of the discharge notes are much longer than the 510, which is the maximum token length that BERT can take at each time during training (Figure 2). In fact only 11.67% of discharge summaries have a length of tokens shorter than 510, so except for the first “bump” (in Figure 2) of discharge summaries which falls below the 510 length requirement, for the rest 90% of the data, only a proportion of the discharge note can be fed into the model, making BERT a bad modeling choice for this particular problem. However, the first 510 tokens still contain meaningful description of symptoms and diagnosis, and there is a new study showing that a simple text truncation (i.e. head+tail) works the best on IMDB data than hierarchical model that stack LSTM or attention on top of BERT to connect different sections of paragraphs [9]. There is also a study that shows breaking the long paragraphs into different samples works well enough for predicting hospital readmission [5]. Given the convenience of the fine-tuning approach and the nature of the project as a fun course project, it may still be of interest to try out BERT and evaluate its performance with tweaks to the data preprocessing, and pin-down the component that might

improve the model performance on ICD code prediction.

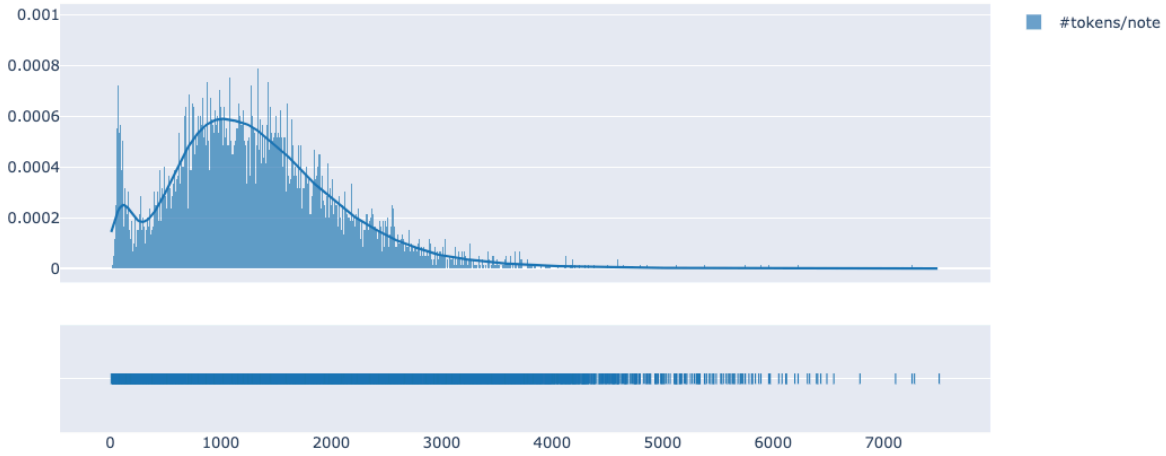

Figure 2 Distribution of the Total Number of Tokens Per Discharge Note

2.4 Preprocesing

Two preprocessing approaches were used, one following Mullenbach and one following Huang. In Mullenbach's study which the researchers used convolutional neural network with attention mechanism, the tokens were "cleaner" in the way that all punctuation marks are removed, along with special characters, and tokens that has no alphabetic characters (e.g., removing "500" but keeping "250mg"). In Ke's study, only special characters are removed and everything else is retained. Both studies use lower-cased tokens. The first approach will make the data directly comparable to previous study results, and also allow the 510-length input to include more "meaningful" word tokens than using the second approach. However, the second approach of preprocessing is also considered because BERT is pretrained on a vocabulary that contains both alphabetic characters and punctuation marks. Although punctuation marks may seem meaningless for convolutional network which is the model Mullenbach used, they may have "meanings" in language modeling and certainly have values to humans such that they help people process information better. It is therefore of interest to see how well the model performs on both types of input data.

2.5 Evaluation Metrics

To facilitate comparison with prior work, I reported both the macro- and micro-averaged area under the ROC curve (AUC) and F1 score. Micro-averaged values are calculated by treating each (text, code) pair as a separate prediction. Macro=averaged values are calculated by averaging metrics computed per-label.

3. Results

As expected, the BERT model's performance is inferior to all the convolutional neural network models which are known to work well for concept extraction tasks over arbitrarily long text sequence (Table 1). Yet although BERT was trained on only on the first chunk of the input sequence, its performance still exceeds Bi-GRU and ULMFiT which presumably should be able to handle text with arbitrary length better than a model that can only take fixed length of data.

Table 1 Comparison of BERT's results to previous studies

	AUC		F1	
Model	Macro	Micro	Macro	Micro
C-MemNN (Prakash et al., 2017) [condensed memory neural network]	0.833	N/A	N/A	N/A
Shi et al. (2017) [neural language model]	N/A	0.9	N/A	0.532
Logistic Regression (Mullenbach, et al.,2018)	0.829	0.864	0.477	0.533
CNN (Mullenbach, et al.,2018)	0.876	0.907	0.576	0.625
Bi-GRU ((Mullenbach, et al.,2018)	0.828	0.868	0.484	0.549
CAML (Mullenbach, et al.,2018)	0.875	0.909	0.532	0.614
DR-CAML((Mullenbach, et al.,2018)	0.884	0.916	0.576	0.633
ULMFiT (Nuthakki et al., 2019)	N/A	N/A	N/A	0.55* 0.48*
BERT-Large	0.858	0.889	0.531	0.605

* F-1 score is computed for diagnosis and procedure codes separately.

When comparing within the current studies on different BERT models with two types of preprocessing, $BERT_{LARGE}$ yields better performance than $BERT_{BASE}$ (Table 2). One interesting contrast is also revealed when comparing across the two preprocessing approaches. When the model is smaller ($BERT_{BASE}$), cleaning the input more aggressively by removing all punctuations helps improving the model performance by about 1% on the AUC or F1. Yet after upgrading to $BERT_{LARGE}$ the difference was gone or even reversed, where input data with punctuation marks performs even better than $BERT_{BASE}$ on the small test set that contains instances with at least one of the top 50 most frequent codes.

Table 2 Results on MIMIC-III, 50 labels, within study comparison

Test on the same subset as Shi et al. (2017) (N=1,730)	AUC		F1	
	Macro	Micro	Macro	Micro
With Puncs				
Bert-base-uncased	0.822	0.864	0.468	0.555
Bert-large-uncased	0.858	0.889	0.531	0.605
Without Puncs				
Bert-base-uncased_512	0.832	0.870	0.476	0.568
Bert-large-uncased_512	0.848	0.883	0.522	0.594
Test on the full test set 50 labels (N=3,372)	AUC		F1	
	Macro	Micro	Macro	Micro
With Puncs				
Bert-base-uncased	0.822	0.864	0.464	0.553
Bert-large-uncased	0.852	0.885	0.519	0.591
Without Puncs				
Bert-base-uncased_512	0.837	0.873	0.472	0.564
Bert-large-uncased_512	0.853	0.886	0.518	0.590

The truncation does appear to affect model performance (Table 3). I compare the frequency of the ICD codes with highest and those with lowest AUC scores on the frequency of the code as well as the key-word (i.e. terms related to the name of the ICD code) frequency in the token

set. Because these ICD codes are already the top-50 most frequent codes, so the frequency of the codes is not too different. The top predicted codes show up 40% or 1.4 times more than the bottom predicted codes. If the entire sequence of discharge notes can be included in the model, we may expect a similar ratio for key-word frequency. Yet when comparing the frequency at the key-word or token level, the frequency of key terminology is 400% or 4.27 times more than the bottom predicted codes. The top predicted codes have more key terminology being included in the first 520 tokens of the discharge notes. Additionally, procedure codes are easier to predict than diagnosis codes.

Table 3 Comparison of ICD codes with top-10 and bottom-10 AUC by frequency of ICD codes and key-word tokens

Code	AUC	Name of the ICD code	Type	Code Frequency (N=52,726)	Key-word Frequency in the first 520 Tokens
Top10-ranked				Total: 51,282	Total: 278,614
36.15	0.9947	Coronary artery bypass	Procedure	3981	76682
39.61	0.9922	Extracorporeal circulation auxiliary	Procedure	6141	339
88.56	0.9661	Coronary arteriography	Procedure	4625	2904
39.95	0.9583	Hemodialysis	Procedure	2889	2366
414.01	0.9536	Coronary atherosclerosis	Diagnosis	10990	27875
37.22	0.9499	Left heart catheterization	Procedure	2861	46814
37.23	0.9491	heart cardiac catheterization	Procedure	2465	79678
V45.81	0.9455	Aortocoronary bypass	Diagnosis	2650	12149
427.31	0.9195	Atrial fibrillation	Diagnosis	11329	25134
995.92	0.9152	Severe sepsis	Diagnosis	3351	4673
Bottom10-ranked				Total: 36,549	Total: 65,231
276.2	0.7557	Acidosis	Diagnosis	3799	1078
99.04	0.743	Transfusion of packed cells	Procedure	6944	7902
599	0.7364	Urinary tract infection,	Diagnosis	5766	5868
276.1	0.7344	Hyposmolality and/or hyponatremia	Diagnosis	2566	971
287.5	0.7337	Thrombocytopenia	Diagnosis	2602	1188
511.9	0.7331	Unspecified pleural effusion	Diagnosis	2413	17945
V15.82	0.7227	Personal history of tobacco use	Diagnosis	2251	15089
305.1	0.7045	Tobacco use disorder	Diagnosis	2775	15089
311	0.7017	Depressive disorder	Diagnosis	2819	101
285.9	0.6659	Anemia, unspecified	Diagnosis	4614	0

4. Discussion

BERT does not seem to be an ideal modeling choice for predicting ICD codes from medical notes due to its ability to handle sequence of fixed length. But its performance still exceeds some of the baseline models which were fed the entire corpus of notes. This may prove that pretrained BERT or transformer stacks having better text feature extraction ability which makes it work better than model with inferior extraction ability despite of shortage of data. It might be of interest for future researchers to modify the structure of BERT to allow some recurrence to handle longer sequence of input. But for this particular task – ICD code

prediction – it may actually be more valuable to improve upon a CNN type of model than a BERT based model given the former is lighter, requires no pretraining, and works well.

I also experimented with splitting the notes into pieces and feed into the model (results not presented). But the model performance actually dropped a bit by doing so. This is not too surprising as each chunk of 510 token may only contain pieces of information related to part of the ICD codes, which actually introduced more "noises" to the model than "information". I did not end up running models on the "head-tail" truncation due to time limitation on this project. But a closer look at the discharge notes reveals that the last part is typically about recovery, while the beginning section is typically symptom description, so the first 510 tokens may be more relevant for reflecting the diagnosis of a patient than the combination of first 255and the last 255 tokens. Yet it is just an assumption. The other things that deterred me from trying is because although various types of truncation may work for certain data, an approach that's too dependent on data-structure may be less robust to a structure-neural model, and could be difficult of scale up when moving from one dataset to another where the data structure or format of discharge notes changes.

For previous studies, the number of outcome classes is typically small, and hence a simple truncation works well enough. But for a task that needs to predict labels over 50 different categories or even thousands of categories in a full ICD code set, missing one sentence could potentially mean the information related to one or several classes are gone forever. This explains why the magic of BERT no longer works for this particular down-stream task. It is actually somewhat surprising to see that the model still have some predictive power despite that 90% of the discharge notes are truncated. This potentially reflects that the first chunk of discharge notes may have contained a decent amount of diagnostic or procedural information.

All prior studies which repeat the pretraining process of BERT on MIMIC-III dataset have only adopted $\text{BERT}_{\text{BASE}}$ (cite biobert and clinicalberts). Yet it turns out $\text{BERT}_{\text{LARGE}}$, despite of its 'ridiculously' large size, which makes it appear susceptible to over-fitting on a relatively smaller MIMIC dataset, it actually ends up performing much better than $\text{BERT}_{\text{BASE}}$ for medical text classification and requires less data preprocessing, thanks to the boosted number of parameters which are pretrained. Future researchers interested in pretraining BERT model on clinical dataset may consider size up the model to $\text{BERT}_{\text{LARGE}}$.

Reference

1. Liendo, Z., G. De Roo, and A. Karmakar. *Classifying medical notes into standard disease codes*. 2018; Available from: https://arxiv.org/pdf/1802.00382.pdf.
2. Nuthakki, S., et al., *Natural language processing of MIMIC-III clinical notes for identifying diagnosis and procedures with neural networks.* arXiv:1912.12397, 2019.
3. Huang, J., C. O'sorio, and L. Wicent Sy, *An empirical evaluation of deep learning for ICD-9 code assignment using MIMIC-III clinical notes.* Computer Methods and Programs in Biomedicine, 2019. **177**: p. 141-153.
4. Mullenbach, J., et al. *Explainable Prediction of Medical Codes from Clinical Text*. in *NAACL-HLT 2018*. 2018.
5. Huang, K., J. Altosaar, and R. Ranganath, *ClinicalBERT: Modeling CLinical Notes and Predicting Hospital Readmission.* arXiv:1904.05342, 2019.
6. Devlin, J., et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. in *NAACL-HLT 2019*. 2019.
7. Alistair E.W. Johnson, et al., *MIMIC-III, a freely accessible critical care database.* Scientific data 3, 2016.
8. Black, S., et al., *Efficacy, safety and immunogenicity of heptavalent pneumococcal conjugate vaccine in children. Northern California Kaiser Permanente Vaccine Study Center Group.* Pediatr Infect Dis J, 2000. **19**(3): p. 187-95.
9. Chi Sun, et al., *How to Fine-Tune BERT for Text Classification?* arXiv, 2020. **1505.04597 [cs.CV]**.