# Medical Code Predictions from Clinical Diagnosis and Procedures Using Natural Language Processing

**Yuanning Zheng, B.S[1], Chao Pu, Ph.D[1] George Zhou, B.S[1], Sichao Jia, M.S[1]**
**[1]Georgia Institution of Technology, Atlanta, GA 30332, USA**

## Abstract

*Annotating clinical descriptions that are free text narratives from clinicians with the diagnostic codes can be challenging. The labeling process is time-consuming and error-prone, especially when dealing with a large volume of clinical records. Therefore, a machine-based prediction of the ICD code is in urgent demand. In this study, we performed conduct multi-label classification predictions for ICD-9 code based on diagnosis summary from MIMIC dataset. We implemented training methods that use three different language models: (1) Deep-neural network-based models that are established on the word2vec CBOW word embedding methods; (2) Models that use skip-gram word embeddings; (3) Clinical Bidirectional Encoder Representations from Transformers (Clinical BERT), a contextual language mre based on unidirectional language modelodel that has been pre-trained on clinical texts. Finally, we compared the performance of these models.*

Slides and video presentation are available here and here, respectively and code is on Github (CAML Models and ClinicalBert ICD Models). Fine-tuned model can be found from here.

## 1 Introduction

Electronic health records, such as those from the MIMIC III database, employ the International Classification of Diseases (ICD) codes to dictate diagnoses and procedures during patient encounters. However, transforming clinical descriptions that are free text narratives to standard ICD codes can be challenging. The labeling process is time-consuming and error-prone, especially when dealing with a large volume of clinical records. Moreover, the annotating process is subjective; the resulting codes are established on the clinician's experience and therefore can vary between different clinicians regarding the same clinical event. An emerging challenge is adapting ICD codes from the old version (ICD-9) to the newest one (ICD-10). Given that the MIMIC III database currently contains more than 40 thousand discharge summaries, it is impractical to reannotate the entire records from the current available ICD-9 to ICD-10 by professional clinicians. Therefore, a machine-based prediction of the ICD code is in urgent demand.

### 1.1 Previous work

Automatic ICD coding methods have been widely investigated utilizing manual rules[1] or feature-based selection approaches[2]. However, the above methods usually involve heavy parameter tuning and human subjects. Additionally, some of these approaches are also constrained to assign merely one single diagnosis code to one diagnosis entry, which naturally excluded some informative content from the clinical texts.

Recent research leveraged machine learning models to extract information from unstructured text data, and defined a multi-label classification problem to predict the ICD code from hospital records[3,4]. *Hasan et al.* presented a condensed memory neural network (C-MemNNs), which exploited raw text from Wikipedia as a knowledge source to predict ICD code from patient discharge notes from the MIMIC III database[8]. The C-MemNNs maximize the utility of the memory slots in the network, which improves the accuracy of diagnostic inferencing. Ayyar *et al*. demonstrate a long short-term memory (LSTM) model with a single layer of nonlinearity that can be used to predict ICD-9 labels efficiently [9].

Recently, convolutional neural network (CNN)-based framework that combines convolution with attention to select the most relevant parts of the discharge summary[3,5,6] were proposed. Both word-based and character-based long short-term memory (LSTM) recurrent network models have been investigated and applied on discharge summaries for medical coding[4]. All aforementioned machine learning frameworks provide robust data-driven approaches for medical coding from a large volume of clinical records. For instance, *Mullenbach et al*. proposed a Convolutional Attention for Multilabel Classification (CAML) framework[3]. CAML aggregates information across the document using a convolutional neural network and uses an attention mechanism to select the most relevant segments for each of the thousands of possible codes.

However, all the aforementioned models are based on context-free word encoding models, such as word2vec or Glov. By generating a single word embedding representation for each word in the vocabulary, these traditional language models are unidirectional. Recently, several contextual models have been developed, such as BERT (Bidirectional Encoder Representations from Transformers). BERT applies bidirectional training of Transformer to language modeling.  It generates word representation based on other words in the sentence that enables it to learn the bidirectional context of the word in sentences. Through a deeper understanding, BERT has been shown to outperform standard models that use single directional word embeddings. Google research has released BERT BASE, pre-trained BERT models using texts from Wikipedia.  In subsequent studies, researchers adapt BERT BASE to multiple domain specific data sets. For instance, researchers trained BioBERT by using texts from biomedical journals and SciBERT trained by scientific text.

## 2. Method

### 2.1 Dataset and preprocessing

The project will be conducted using the public MIMIC-III dataset, which contains the de-identified health records of over 40,000 patients who stayed in critical care units of Beth Israel Deaconess Medical Center between 2001 and 2012[7].

Following the processing work introduced in J. Mullenbach *et al*[3], we leverage the usage of PySpark - the Python API for Spark framework in the data ETL pipeline. We concatenated the records from DIAGNOSES_ICD and PROCEDURES_ICD tables and reformatted the ICD-9 codes to normal conventions. These ICD-9 codes were tagged by human coders previously, and there were 8,994 unique codes in the whole set[4]. Then, we extracted the discharge summary record from the DIAGNOSES_ICD table, and tokenized the free format raw text data in the TEXT column with RegexpTokenizer. Tokens with no alphabetic characters was removed. All tokens will be in lowercase. Tokens that appear in less than 3 documents will be replaced with a ‘UNK’ token.

In the next step, we joined the tokenized discharge summary with the ICD-9 code tables extracted earlier based on matching SUBJECT_ID and HADM_ID combination. Since it is possible that some patients will have multiple visits, the training and testing samples will be aggregated by patient id. The record entries with no discharge diagnosis will be discarded.

The distribution of token size in both training and testing datasets were plotted in Fig. 1. The median token size is 1341 for training data and 1683 for testing data. The label count is not strongly correlated with text token size, but we can still observe the positive correlations between these two variables.

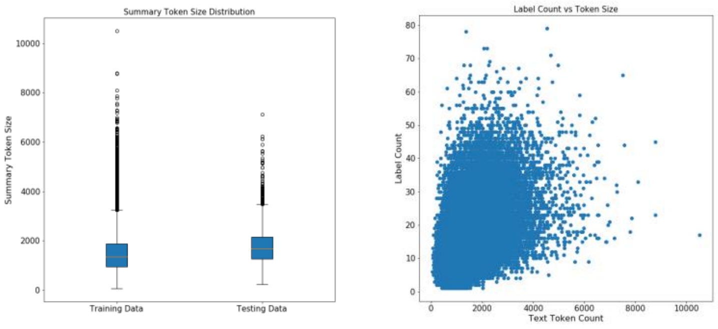

Figure 1. (a) Discharge summary token size distribution (b) correlation between label count and text token count

### 2.2 Architecture

In this section, we demonstrate the architecture and implementation of the CAML model and the BERT model.

### 2.2.1 Implementation of CAML

We utilized the Word2Vec Continuous Bag of Words (CBOW) method introduced in Mikolov *et al*[10] and produced the word embeddings from a large corpus of text from the discharge summary. CBOW method uses a surrounding contextual word (a.k.a bag of words assumption) to predict one word with a shallow layered neural network. In contrast, another common Word2Vec method, Skip Gram, is more robust for infrequent and distant words. Both methods were investigated in this study, and the results are compared in a later section.

The dimensionality of the label space of the ICD-9 code provides abundant possibilities of the diagnosis. Therefore, many codes have not been covered by the available data. To tackle this problem, we will use text descriptions of each code from the World Health Organization (2016) to implement a secondary module in our network. This secondary module can produce regularizers of the model parameters if a specific code did not appear in the MIMIC III database.

The architecture of CAML has been described previously[3]. Briefly, the model takes the list of word vectors of each instance as input. The input matrix X had a size of $(d_e, N)$, where $d_e$ is the embedding dimension of the word and N is the sequence length.  This input will then go through a convolution layer, which outputs a matrix H with a size of $(d_c, N)$, where $d_c$ is the size of the convolution filter output. Then the attention mechanism is applied by multiplying the $H^T$ by the vector parameters $(u_l)$ of each label l, which produces the attention vector with the size of (N, Y) , where Y is the label space. The attention vector represents the distribution of each label in the text, which will then be multiplied by the matrix H to achieve a vector representation $V_l$ of each label, which has a size of $(d_c, Y)$. A linear layer and sigmoid function will then be applied to V to achieve a vector that represents the probability of each label. The overall architecture is described in **Figure 2**.

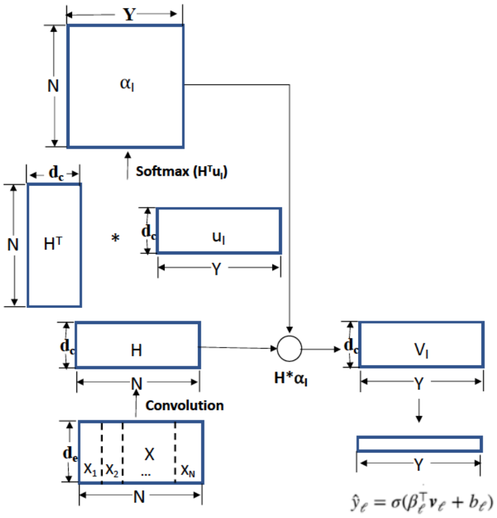

**Figure 2. CAML architecture with per-label attention**

### 2.2.2 Implementation of BERT

Recently, Alsentzer et al. trained Clinical BERT using clinical texts from the MIMIC notes. They released four BERT models trained by clinical texts of different sources. Clinical BERT or Clinical BioBERT is trained on all types of notes from MIMIC database. They are initialized from BERT BASE and BioBERT, respectively. Whereas, the Discharge Summary BERT `and the Bio +Discharge Summary BERT are trained on discharge summaries. The results showed that these models outperform the standard BERT in specific medical NLP tasks. The model was downloaded from the git repository from the original paper on ClinicalBERT. We adapted the *BertForSequenceClassification* class from the Huggingface's Pytorch implementation of BERT. The model implemented a dropout layer following the output from the BERT layer. A linear layer was then added on top of the dropped layer for classification purpose. The original example was to address the binary classification task. To adapt the code for addressing the multilabel classification problem, we used the Binary cross-entropy with logits method to calculate the loss, instead of the original vanilla cross-entropy loss.

### 2.2.3 Training

The training parameters were similar to the original BERT implementation in run classifier example. We trained the model for 4 epochs with a batch size of 10 and maximum sequence length as 512. The learning rate was 3e-5 as recommended in the original paper. The model was trained on Google Cloud Platform with a n1-standard-1 virtual machine equipped with 1 NVDIA Tesla V100 GPU.

## 3 Results

ETL and data preprocessing was conducted using Apache Spark. To train the deep learning neural network, we employed Python with PyTorch library.  The algorithms that were used for code annotation have been described in the above. In terms of hardware, we initially used local machines to run the ETL and model training. In order to speed up the training process, we took advantage of Nvidia Tesla K80 and V100 GPU's on Google Cloud Platform (GCP).

As shown in Table 1 and 2, we started our study by reproducing the results by using CAML proposed by J. Mullenbach *et al*. The results for CAML-50 - where 50 means the top 50 most frequent codes, CAML-full - where full means all codes, and DRCAML-full were generated using a pretrained model provided in the GitHub repository. If the training script was provided, the other algorithms tested in the paper -i.e. logistic regression, vanilla-CNN, Bi-GRU - were trained using the default hyperparameter settings.Besides reproducing the original study, we added additional models using datasets trained using Word2Vec Skip Gram approach rather than Word2Vec Continuous Bag of Words (CBOW) approach. Skip Gram approach attempts to map a single context word to neighboring target words, whereas CBOW approach attempts to map neighboring context words to a single target word. As such, we assumed skip-gram approach would perform worse for P@n - i.e., precision for top n highest scored labels that are present in the ground truth - and better at predicting more rare words. As the original paper by J. Mullenbach *et al* does not mention the Skip Gram approach, testing the MIMIC-III dataset using the Skip Gram approach helps to verify that, for one, that CBOW performs better, and secondly, that the dataset we are working is not very large as skip-gram tends to perform better on larger datasets.

**Table 1.** Evaluation metrics of different models

| **Top 50 codes – Test set** | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **Accuracy** | | **Precision** | | **Recall** | | **F-score** | | **AUC** | | **P@5** | |
| | macro | micro | macro | micro | macro | micro | macro | micro | macro | micro | - | |
| CAML | 0.393 | 0.443 | 0.604 | 0.714 | 0.480 | 0.538 | 0.535 | 0.614 | 0.877 | 0.910 | 0.611 | |
| DRCAML | 0.368 | 0.423 | 0.578 | 0.716 | 0.446 | 0.509 | 0.503 | 0.598 | 0.862 | 0.905 | 0.599 | |
| CNN | 0.430 | 0.461 | 0.520 | 0.565 | 0.670 | 0.713 | 0.586 | 0.631 | 0.879 | 0.909 | 0.617 | |
| Clinical Bert | 0.145 | 0.214 | 0.416 | 0.727 | 0.165 | 0.233 | 0.236 | 0.353 | 0.782 | 0.828 | 0.471 | |
| **Full dataset – Test set** | | | | | | | | | | | | |
| | **Accuracy** | | **Precision** | | **Recall** | | **F-score** | | **AUC** | | **P@n** | |
| | macro | micro | macro | micro | macro | micro | macro | micro | macro | micro | 8 | 15 |
| CAML | 0.061 | 0.369 | 0.091 | 0.607 | 0.086 | 0.484 | 0.088 | 0.539 | 0.895 | 0.986 | 0.709 | 0.561 |
| DRCAML | 0.059 | 0.360 | 0.085 | 0.553 | 0.088 | 0.508 | 0.086 | 0.529 | 0.897 | 0.985 | 0.698 | 0.546 |
| Bi-GRU | 0.028 | 0.279 | 0.057 | 0.584 | 0.038 | 0.349 | 0.045 | 0.436 | 0.846 | 0.975 | 0.619 | 0.469 |
| CNN | 0.024 | 0.271 | 0.049 | 0.550 | 0.035 | 0.348 | 0.040 | 0.426 | 0.814 | 0.970 | 0.594 | 0.454 |
| Clinical Bert | 0.001 | 0.050 | 0.001 | 0.052 | 0.02 | 0.54 | 0.002 | 0.095 | 0.525 | 0.940 | 0.225 | 0.194 |

**Table 2.** Evaluated metrics for selected models with Skip-gram Word2Vec method

| **Top 50 codes -Test set** | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **Accuracy** | | **Precision** | | **Recall** | | **F-score** | | **AUC** | | **P@5** | |
| | macro | micro | macro | micro | macro | micro | macro | micro | macro | micro | - | |
| CAML | 0.358 | 0.390 | 0.570 | 0.630 | 0.453 | 0.505 | 0.505 | 0.561 | 0.853 | 0.878 | 0.554 | |
| DRCAML | 0.368 | 0.424 | 0.588 | 0.717 | 0.445 | 0.509 | 0.507 | 0.595 | 0.863 | 0.904 | 0.598 | |
| **Full dataset - Test set** | | | | | | | | | | | | |
| | **Accuracy** | | **Precision** | | **Recall** | | **F-score** | | **AUC** | | **P@n** | |
| | macro | micro | macro | micro | macro | micro | macro | micro | macro | micro | 8 | 15 |
| CAML | 0.046 | 0.348 | 0.076 | 0.637 | 0.061 | 0.434 | 0.068 | 0.516 | 0.886 | 0.984 | .695 | 0.545 |
| DRCAML | 0.040 | 0.340 | 0.069 | 0.634 | 0.052 | 0.424 | 0.059 | 0.508 | 0.877 | .984 | .684 | 0.542 |

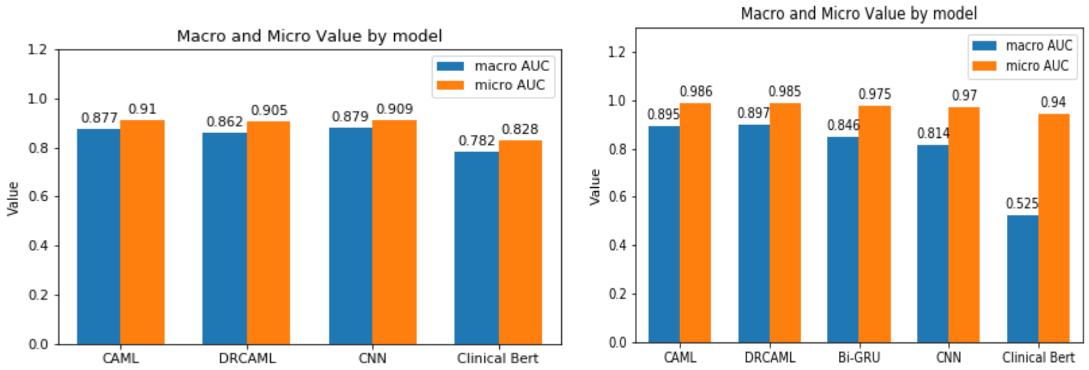

**Figure 3.** Evaluation metrics comparison for different model with (left) top 50 ICD codes and (right) full ICD code set using CBOW Word2Vec method

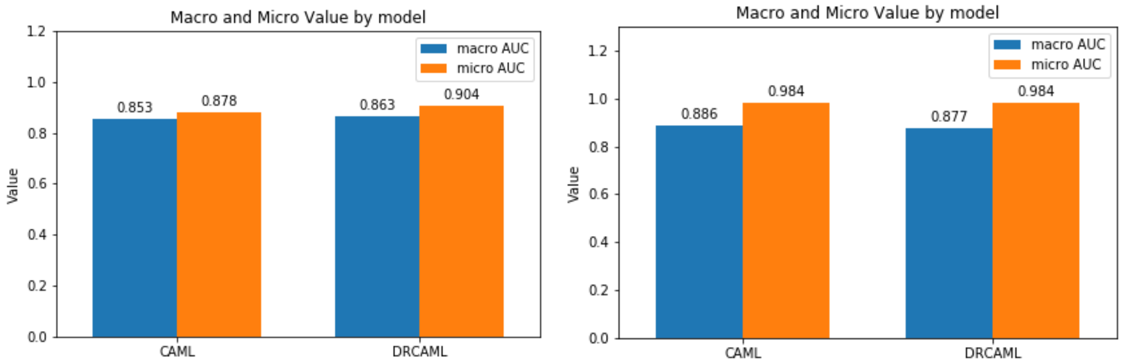

**Figure 3.** Evaluation metrics comparison for different model with (left) top 50 ICD codes and (right) full ICD code set using Skip-gram Word2Vec method

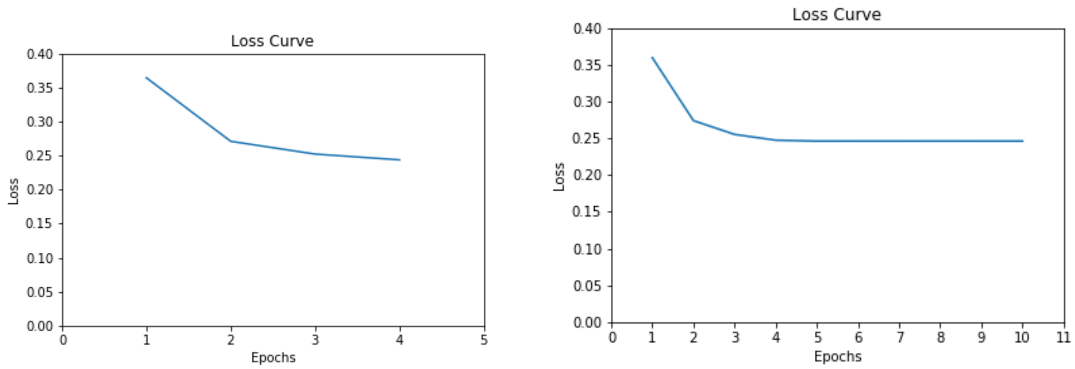

**Figure 4.** Lost vs Epoch on training top 50 icd codes (Left: 4 epoches. Right: 10 epoches)

## 4. Discussion

In order to improve the results on the MIMIC-III dataset, we intend to develop a model using Bidirectional Encoder Representation from Transformers (BERT) developed by GoogleAI. BERT has been shown to perform very well on 11 different NLP tasks in the General Language Understanding Evaluation (GLUE) benchmark. BERT uses transformers to generate encoder representations for words from either side of a sentence. The intuition behind this approach lies in the fact that the prediction of a word has a higher probability of success if given the context of the word - i.e. by considering text from either side of the word. BERT uses multi-headed attention blocks to focus on specific parts of text that it deems to be the most important.

We used a pre-trained BERT-like model and will fine-tune it to our specific dataset. There have been many models inspired by BERT (XLNet, Distilbert, RoBERT, ALBERT, ERNIE 2.0, etc.). One model, A Lite BERT (ALBERT), also by GoogleAI, has been shown on GLUE benchmark to perform even better than BERT while requiring about 18x less parameters with faster training time. With a faster and smaller model, we could fine-tune our models more with less hardware restrictions. Future work could include exploring ALBERT for clinical datasets.

Finally, to further justify our rationale for using BERT, unlike CAML model which is a left-to-right language for pre-training, BERT is pre-trained bi-directionally for language representations by using two unsupervised tasks[11,12]. The first task is Masked language model which simply mask some percentage of the input tokens at random, and then predict those masked tokens[12]. Another drawback is that CAML model did not capture the relationship between two sentences, whereas ICD-9 code prediction heavily relied on the comprehensive understanding of information distributed in nearby sentences in clinical notes. Therefore, Next Sentence Prediction, as the second pre-training task of BERT, may solve this problem by training a sentence's next sentence with the actual next sentence at 50% of the time or a random sentence from the corpus at another 50% of the time[12]. The author stated that this simple model for this task was very beneficial to both Question Answering and Natural Language Interaction, two fairly popular NLP tasks[12]. Therefore, we predicted that BERT model could also be useful for ICD-9 code prediction from the clinical notes.

A major reason why BERT failed to improve the prediction performance is that the model restricts the maximum input sequence length to 512 tokens. We found that the median length of the input sequence was over 1341 tokens. Therefore, at least half of the information was lost in half of our training data. The effect of this input limitation was magnified by the large labeling space. The information for predicting a label can either be totally lost, resulting in a miss of prediction, or be partly truncated, diminishing the resolution of predicting similar labels.

A possible solution to tackle this problem is to split the long texts into pieces of short subtexts. An example of this is the sliding window approach used in the original PyTorch implementation of running the SQuAD example. Oversized input messages are split into subtexts which are fed into the model as separate instances. The output logits will then be combined using sum or selection of the max value from different predictions of the same sample. This approach can potentially improve the model performance, but it will increase the computational expenses of the training process.

## 5 Conclusion

We reproduced the ICD code prediction tasks by reproducing the CAML model[3]. Following the first phase of this work, we extended the prediction task with the latest state-of-art BERT model, intending to improve the prediction performance. The original ICD code mutli-labeling tasks were adapted to a pre-trained clinical specific BERT model proposed by Alsentzer, E., *et al.* Next, this clinical specific BERT model was then fine-tuned with MIMIC3 discharge summaries. However, the evaluation metrics indicated a worse performance comparing with the initial CAML model. Because of the maximum length limitation (512 text length) from BERT model, a majority of text information was truncated automatically after maximum sequence limit. That means BERT model naturally suffered from information loss during both training and testing. In this case, an advanced methodology needs to be explored to fit the long text into BERT model, which could be the principal target of future work.

## References

1. Franz P, Zaiss A, Schulz S, Hahn U, Klar R. Automated Coding of Diagnoses - Three Methods Compared. In: Proceedings of the AMIA Symposium.; 2000:250-254.
2. Scheurwegs E, Cule B, Luyckx K, Luyten L, Daelemans W. Selecting relevant features from the electronic health record for clinical code prediction. J Biomed Inform. 2017;74(2017):92-103.
3. Mullenbach J, Wiegreffe S, Duke J, Sun J, Eisenstein J. Explainable Prediction of Medical Codes from Clinical Text. 2018:1101-1111.
4. Shi H, Xie P, Hu Z, Zhang M, Xing EP. Towards Automated ICD Coding Using Deep Learning. arXiv:171104075. 2017:1-11.

5. Yang Z, Yang D, Dyer C, He X, Smola A, Hovy E. Hierarchical attention networks for document classification. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. ; 2016:1480-1489.
6. Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate. arXiv Preprint arXiv14090473. 2014.
7. Johnson AEW, Pollard TJ, Shen L, et al. MIMIC-III, a freely accessible critical care database. Sci data. 2016;3:160035.
8. Prakash A, Zhao S, Hasan SA, Datla V, Lee K, Qadir A, et al. Condensed memory networks for clinical diagnostic inferencing. 31st AAAI Conf. Artif. Intell. AAAI 2017, 2017.
9. Ayyar, Sandeep, O. B. Don, and W. Iv. "Tagging patient notes with icd-9 codes." In Proceedings of the 29th Conference on Neural Information Processing Systems. 2016.
10. Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, Jeff Dean. Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems. 2013; 3111–3119.
11. Mullenbach, J., Wiegreffe, S., Duke, J., Sun, J., & Eisenstein, J. (2018). Explainable prediction of medical codes from clinical text. ACL Anthology 2018 June: 1101-1111
12. Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2019;1:4171–4186
13. Alsentzer, E., Murphy, J. R., Boag, W., Weng, W. H., Jin, D., Naumann, T., & McDermott, M. (2019). Publicly available clinical BERT embeddings. *arXiv preprint arXiv:1904.03323*.

## Appendix. Team Contribution

1. Yuanning Zheng worked on implementing code for model training and evaluation locally and provided images/description for CAML model.
2. Chao Pu mainly worked on implementing the code for model training and evaluation at Google Cloud platform. Additionally, he worked on the refactoring the data pre-processing with PySpark package to leverage the usage of big data tools.
3. George Zhou wrote and tested skip-gram comparison and ran the all models used for comparison with our BERT model. He additionally worked on other preprocessing techniques that were excluded from paper. He also created video/presentation.
4. Sichao Jia generated analysis graphs and prepared papers for draft and final paper. He also brought findings from outside resources to our paper.