Predicting ICD-9 codes based on clinical notes from MIMIC-III dataset using Deep Neural Networks

Ziyu Qiu
Cornell University
New York, NY 10044
zq64@cornell.edu

Ta-Wei Mao
Cornell University
New York, NY 10044
tm592@cornell.edu

Jingxuan Sun
Cornell University
New York, NY 10044
js3422@cornell.edu

Yezhou Ma
Cornell University
New York, NY 10044
ym462@cornell.edu

Yixue Wang
Cornell University
New York, NY 10044
yw2224@cornell.edu

Abstract

Automatic mapping medical notes to diagnose codes can remarkably boost the efficiency of clinicians and make medical billing process more seamless and reliable. However, this task is rather challenging due to complicated medical note formats and codes. Benchmarked previous researches, we inspect the model performance of Hierarchical Attention Network (HAN), Hierarchical Self-Attention Network, Codes Attentive Long-Short Term Memory (LSTM)[1], Self-Attention LSTM and fine tuning on the pre-trained Clicnical BERT using MIMIC-III dataset.

1 Introduction

Medical coding is an essential part of the medical billing process. After a patient is seen, the care provider needs to fill out a billing form with the corresponding medical codes (disease or procedure) to collect payments from the insurance companies or government.

Currently, this process of converting the free-text medical notes to standardized medical codes is mostly done manually either by clinicians themselves, or professional medical coders. However, manual coding is time-consuming and error-prone. Once the bill gets rejected, it is tough for whoever involved in the case to recall what happened three to six months ago, so it became accounts receivable that is highly likely not collectable in the future. As a result, getting the notes coded accurately and efficiently becomes a hot area in recent years. Researches on automatic coding emerged since the 90s.

One such task is to learn a mapping from natural language free-texts to medical concepts such that, given a new document, the system can assign one or more International Statistical Classification of Diseases (ICD) codes to the clinical notes. The task is inherently challenging for a few reasons. (i) the label space is very high-dimensional, there are 8,921 unique ICD-9 codes in total (ii) each instance corresponds to multiple labels (up to 38 labels) (iii) instances are long documents that include misspellings, non-standard abbreviations, and medical specific vocabulary (iv) information extraction requires contextual understanding, and critical information to specify sub-codes might be missing.

In this paper, we investigate how to approximating the mapping from the medical notes to predict the medical codes in this setting. This can be achieved through multi-label classification, hierarchical classification, and information retrieval. In our work, We will implement 5 models: (i) Hierarchical Self Attention Network (ii) Self-Attention Long-Short Term Memory (SLSTM) (iii) Codes Attentive

32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.

Long-Short Term Memory (CLSTM) (iv) Hierarchical Attention Network (HAN) (v) Fine Tuning on Pre-trained Clinical BERT[2].

Particularly, we try two types of attention mechanisms for different model architectures. All our models are validated on the publicly available Medical Information Mart for Intensive Care III (MIMIC-III) dataset using macro-F1, micro-F1 and precision@N metrics.

2 Related Work

Automatic assignment of ICD codes to clinical and health related documents has been well studied with different approaches from rule based and dictionary look ups to machine learning models. Recently the focus has been moved to applying deep learning models.

The paper[3] has worked on CLEF eHealth 2019 Task 1[1], multilingual information extraction from German non-technical summaries (NTSs) of animal experiments collected from AnimalTestInfo database to classify according to ICD-10 codes. Several models and pre-trained embeddings were experimented in this work, starting with traditional bag-of-words support vector machines (SVM) as the base line, following experimented with standard deep learning architecture of CNN and recurrent neural networks (RNN) with three types of attention mechanisms: hierarchical attention Gated Recurrent Unit (GRU), self-attention LSTM and codes attentive LSTM. Lastly, it is shown that transfer learning with pre-trained Bidirectional Encoder Representations from Transformers (BERT)[4] models and its recent variant BioBERT perform the best with domain specific embeddings instead of contextual word embeddings.

[5] leveraged the unstructured portion of the Electronic Health Record (EHR), clinical notes, with three characteristics: a very large label set (6,500 unique ICD-9 codes and 1,047 3-digit unique ICD-9 codes), a multi-label setting (up to 20 labels per instance), instances that are long documents (on average 1,900-word long). Four models were investigated for the task of extreme multi-label classification: SVM, Continuous-Bag-Of-Words(CBOW), CNN, and Hierarchical Attention-bidirectional Gated Recurrent Unit model(HA-GRU). The sentence-level tokenization was chosen due to difficulties of training GRU. The results show that with careful tokenization of the input texts along with hierarchical segmentation of the original document, Hierarchical Attention GRU architecture would yield the most promising results with F_1 score of 55.86%, over the SVM, CBOW, and CNN models.

[6] aims to extract ICD-10 codes for cause of death from free text. The authors utilized a variety of structured data sources, including autopsy reports, death certificates and clinical bulletins in Portuguese. The main component of the model is similar to that of [3], which is a hierarchical attention GRU. Due to the characteristics of the reports, the two levels of GRU layers are used to process and represent “word level” and “field level” information. Attention is added for both levels when constructing the final representation. Beyond this, there are several innovative implementations, including setting up 3 separate classifiers on top of the GRUs to classify chapters and labels with sigmoid nonlinearity. The authors also leverage the leveled classification to carry out a fine-grained evaluation. To initialize softmax classifier weights, code co-occurrence matrix is used which is extracted through apriori algorithm. Evaluating on 6 different neural network architectures, the best ensemble model achieved macro-averaged F1-scores of 89.251% or 80.573%.

[7] uses a more traditional dataset, MIMIC-III, for multilabel classification task with Convolutional Attention for Multi-Label classification (CAML). A word embedding is learned through text descriptions of each code from WHO (2016). A convolutional layer is used as a filter on top of a horizontally concatenated word embedding matrix representation of each document. The authors apply an attention mechanism to select the parts of the document that are most relevant for each possible code whose weights are them applied to the base representation. And the result is passed through an output layer with a sigmoid transformation along with a regularizer. The method is accurate, achieving precision@8 of 0.71 and a Micro-F1 of 0.54. Besides, the attention mechanism is proved to be able to identify meaningful explanations with interpretability evaluation by a physician.

In this paper, we conduct multiple experiments using data process approaches and models inspired by the above related works and evaluate the model performance on MIMIC-III dataset.

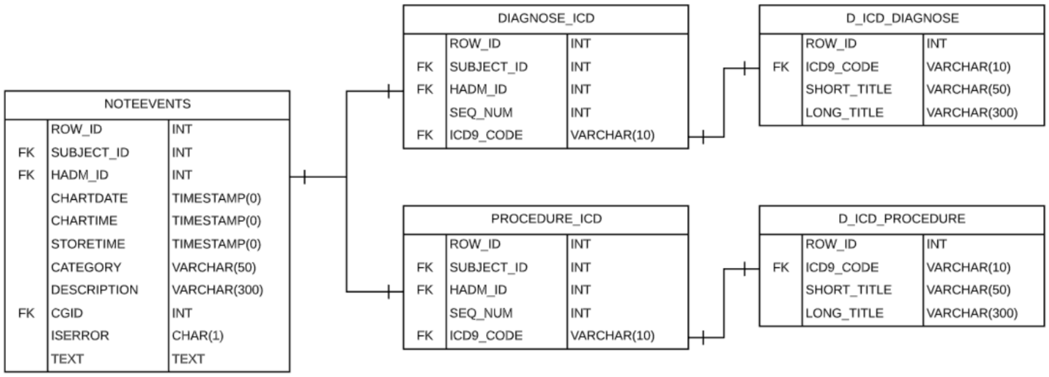
[1] https://clefehealth.imag.fr/?page_id=171

3 Data Processing

MIMIC-III Dataset We will evaluate our approach on MIMIC-III (Medical Information Mart for Intensive Care III). It is an open-access dataset comprising de-identified medical records from Beth Israel Deaconess Medical Center from 2001 to 2012 [8]. The data is associated with 58,976 distinct hospital admissions from 46,520 patients. Each record describes the diagnoses and procedures during a patient's stay, including basic structured information, free-text clinical notes, and ICD-9 codes tagged by humans.

ICD-9 ICD-9 (International Classification of Diseases Version 9), maintained by the World Health Organization (WHO), is a current diagnosis coding system that classifies diseases. Every disease has a unique code in the format of "XXX.XX". Each code contains an alphabetic or numeric first character, two to five numeric digits, and a decimal point between the third and fourth digit [9]. Digits 1-3 indicate the category, and digit 4-5 indicate etiology, specifically anatomic site and manifestations. Examples of ICD-9 codes are: 002.0 (Typhoid fever); 530.81 (Gastro reflux disease); V40 (Mental and behavioral problems). There are a total of 8,921 unique ICD-9 codes in the MIMIC III dataset.

ICD-9 codes are of significant use, including the design of reimbursement systems, the measurement of medical care qualities and the identification of health risks. However, the manual coding process is expensive and time-consuming thus needs to be improved by automatic computational methods. In the MIMIC-III dataset, these codes are assigned at the end of the patient's stay according to their clinical records. By using MIMIC-III as our dataset, we will be able to train a model that effectively learns the mapping from a medical record to its corresponding ICD-9 codes.



	NOTEEVENTS	
	ROW_ID	INT
FK	SUBJECT_ID	INT
FK	HADM_ID	INT
	CHARTDATE	TIMESTAMP(0)
	CHARTTIME	TIMESTAMP(0)
	STORETIME	TIMESTAMP(0)
	CATEGORY	VARCHAR(50)
	DESCRIPTION	VARCHAR(300)
FK	CGID	INT
	ISERROR	CHAR(1)
	TEXT	TEXT

	DIAGNOSE_ICD	
	ROW_ID	INT
FK	SUBJECT_ID	INT
FK	HADM_ID	INT
	SEQ_NUM	INT
FK	ICD9_CODE	VARCHAR(10)

	D_ICD_DIAGNOSE	
	ROW_ID	INT
FK	ICD9_CODE	VARCHAR(10)
	SHORT_TITLE	VARCHAR(50)
	LONG_TITLE	VARCHAR(300)

	PROCEDURE_ICD	
	ROW_ID	INT
FK	SUBJECT_ID	INT
FK	HADM_ID	INT
	SEQ_NUM	INT
FK	ICD9_CODE	VARCHAR(10)

	D_ICD_PROCEDURE	
	ROW_ID	INT
FK	ICD9_CODE	VARCHAR(10)
	SHORT_TITLE	VARCHAR(50)
	LONG_TITLE	VARCHAR(300)

Data Tables For all the data we have, there are 26 tables in total and we chose the most five useful tables as shown above in order to train our model more effectively and efficiently. The labels that we used is ICD9_CODE in DIAGNOSE_ICD. As for the note data, we used TEXT in NOTEEVENTS. The definition of each code is explained in D_ICD_DIAGNOSE.

Input Text During the data processing process, we realize that non of the ID is unique to the medical notes. There are 2,083,180 entries of note taken, yet only 58,362 unique HADM_ID (unique to a patient hospital stay) and 46,146 unique SUBJECT_ID(unique to a patient). It is inferred that there can be multiple records for a patient's admission. In order to match the notes with the labels, we decided to group the notes and codes both by HADM_ID and join them by this key. HADM_ID is preferred over SUBJECT_ID because a single patient might have multiple hospital stays.

As we mentioned earlier, one of the biggest problems for medical data is that they were all recorded manually either by clinicians or professional medical coders, leading to nonuniform format and messy texts containing noises and irrelevant data. We first conducted basic NLP preprocessing procedures, which include stop words removal and special characters removal (Table 1). In addition, we also used NLP preprocess package, *en_core_sci_md*, from *scispaCy* to process biomedical, scientific and clinical text to filter out irrelevant texts and extract more useful vocabularies.

Note Sections The removal and preprocess procedures are still not enough to make the long raw data fit into the models that we are going to train. Another characteristic of the data is that the notes contain different sections recorded by medical professionals such as Procedure, Past Medical

Table 1: Characters Removed

Characters Removed	Interpretation
——	Section Divider
***#	ID
** : ** AM/PM	Time
[** ... **]	De-identified Sensitive Information
\n	New Line

Histories etc., which implies that the raw text were already categorized in a different form. We then used this characteristic to further filter out redundant texts from very common and meaningless sections(Table 2). We manually checked what sections should be discarded in order to extract more meaningful text and make the text length short enough to fit in the models.

Table 2: Sections Removed

Section Headers	Categories Interpretation
Admission Date, Discharge Date, Date of Birth, Date/Time	Time
Completed by, Dictated By, Attending, Provider, MD, PHD,	Person in charge
II, III, IV, VI, VII, X	Roman Numbers
JOB#, # Code	ID

For word tokenization and vectorization, we constructed a dictionary which contains 10000 vocabularies of top frequency among the corpus. If the vocabulary is not in the corpus, it would be labeled as <UNK>(unknown). The length of medical notes after all cleaning procedures ranges from 8 to 342,564, the average value is 4,930 and median is 2,401. Due to the limitation of available GPU memory, we pick 512 (also the maximum sequence length allowed by BERT) for fine-tuning on ClicnicalBERT and 256 for LSTM. Particularly, for HAN model, we pick the first 25 words from the first 10 sentences, resulting in its max sequence length equals 250. If the final length of the processed text doesn't meet the model's required input length, we then padded <PAD> character at the end of the vector.

Output Codes There are 6,985 unique billable ICD-9 codes appeared as labels. As mentioned in *Input Text* section, we concatenated all the ICD-9 codes that are associated with the same HADM_ID(hospital visit) in order to better relate the corresponding notes. However, the model performance predicting the detailed diagnose codes is frustrating, though much better than baseline (details are illustrated in *Evaluation* section).We then try codes at a higher level of hierarchy. As we mentioned before, the code is in the format of "XXX.XX". We only keep the first three digits before the decimal point, which indicates the category. In this way, we decrease the label category to 847 categories. And we further condensed them into 19 groups based on the widely accepted rules[2].

4 Network Architecture

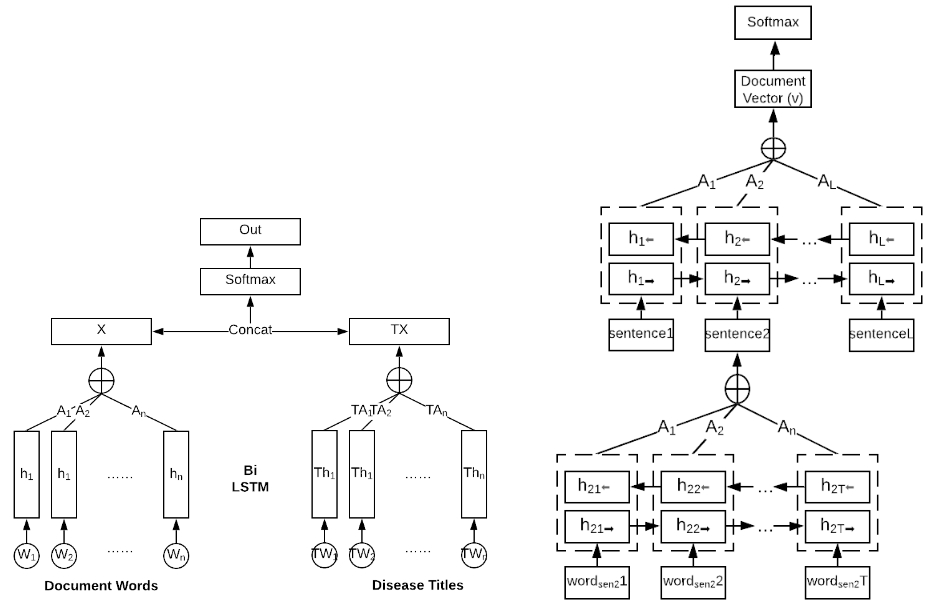
4.1 LSTM

Experiments are first carried out through two varieties of Long-Short Term Memory networks from [3]. The authors used Self-Attention Long-Short Term Memory (SLSTM) and Codes Attentive Long-Short Term Memory (CLSTM) respectively to work on CLEF eHealth 2019 Task 11, multilingual information extraction from German non-technical summaries (NTSs) of animal experiments collected from AnimalTestInfodatabase to predict ICD-10 codes.

For model architecture, both are comparatively shallow networks with one core bi-directional LSTM layer for word-level encoding. Attention is used to generate a probability distribution over features, allowing models to put more weight on relevant features.

Self-Attention LSTM used matrix-matrix multiplication on the input data from the embedding layer to create the attention scores. The input augmented with the self-attention is then passed onto trailing

[2] https://en.wikipedia.org/wiki/List_of_ICD-9_codes



(a) LSTM (b) HAN

Figure 1: Model Architecture

dropout and linear layers. Codes Attentive LSTM takes one step further, using an additional title matrix as input and title embedding matrix that represents the words that appear in the disease titles corresponding to the ICD codes. Such words may or may not be present but the intuition is to use this additional meta-data to enrich the encoder representation by attention. For example, on the highest level, "E000-E999" in ICD-9 codes refers to the category of disease named "Supplementary Classification Of External Causes Of Injury And Poisoning". These vocabulary of the disease are extracted and their embeddings are further retrieved to form the title embedding matrix. Title-attention score is calculated in the same way as self-attention but on the title matrix of the document. For CLSTM, both self-attention and title-attention are applied.

The training is focused on CLSTM as in initial experiments it reached superior performance than the simpler SLSTM. Although [3] used ICD-10 codes which is a more nuanced codec with different grouping criteria, the architectures are suitable for transfer learning on a similar multilabel classification problem on ICD-9. Three types of embeddings are used in the experiments: random initialization, pretrained static word2vec[10] word embeddings, and pretrained dynamic Clinical BERT word embeddings. For CLSTM, different levels of ICD code title embeddings are used according to the Data Processing section. Both networks are trained on a maximum sequence length of 256 words per document with padding.

4.2 HAN

Hierarchical Attention Network (HAN) is another suitable choice to deal with long document classification problems. It applies attention mechanisms at each hierarchical level of a document [11]. As shown in Figure 1(b), an RNN-based word sequence encoder along with a word-level attention layer first encodes each sentence into a vector. Then the sentence vectors are fed into a sentence encoder with a sentence-level attention layer to generate a document vector. This document representation is used by the last fully-connected layer with its corresponding activation function for the final classification. This allows the model to progressively build a document vector by attending and aggregating encoders' outputs at different levels, so as to better capture the hierarchical structure of documents and differentiate importance between contents.

Following this architecture, we use stacked bidirectional Gated Recurrent Units (GRUs) to generate word-level, sentence-level and paragraph-level encoders for the diagnosis descriptions. Encoders are also trained with all three above-mentioned word embeddings (random initialization, pretrained static word2vec word embeddings, pretrained dynamic ClinicBERT word embeddings). Experiments are conducted with both attention and self-attention mechanisms. The maximum number of words in a sentence and maximum number of sentences in a document and are set as 10 and 25 respectively. Sentences in a description are either truncated or padded, and then concatenated together into a sequence of length 250.

4.3 Bert

BERT[4], Bidirectional Encoder Representations from Transformers, is designed to pre-train a deep language representation model which can be easily fine-tuned with one additional output layer to solve the challenges of question answering and language inference etc. Many experiments proved that transfer learning on pre-trained BERT can achieve state-of-the-art performance in the cases using general corpus. Due to the distinct word distribution in biomedical domains, BioBERT[12] is introduced to especially conquer biomedical text mining tasks. ClinicalBERT[2] takes one more step to publish clinical domain specific models using either all MIMIC notes or only its discharge summaries.

Our model is fine-tuned on Clinical BERT using all MIMIC notes version based on BioBERT-Base v1. + PubMed 200K + PMC 270K. We use its default setting as 28996 vocabulary size, 768 embedding dimension and the hidden state dimension is the same as 768. The pre-trained model ends with a BertPooler layer and is concatenated with a dense layer to output logits for multi-label prediction. Sigmoid activation function is leveraged to output probabilities for each code label based on the logits. Only those labels of probability greater than 0.5 would be considered positive. Since the exponential function in sigmoid can cause overflow issue, we apply gradient clipping to the logits before feeding it into the loss function. The clip norm is set as [-0.14, 0.14].

torch.nn.MultiLabelSoftMarginLoss is first used to compute the loss, yet ending in all probabilities lower than 0.5 and thus no positive predictions. By inspecting into the probabilities, we can see more frequently occurred codes tend to have higher probability. This distribution is a common long tailed one causing low recall rate for less common labels.

```
[0.26743987 0.24318083 0.25060663 ... 0.02460566 0.02275323 0.0243159 ]
```

A weight of positive examples is then added to alleviate the unbalanced distribution issue. The *pos_weight* is computed as number of negative examples divided by number of positive ones for each class. Another weighting approach can be leaning to those samples having fewer labels which means considering samples of many labels as “noises". The latter proved to be unsuitable for our tasks, easily causing number overflow issue in training.

NLP models are relatively more sensitive than other models such as computer vision. Small learning rate is required for an effective training. After multiple experiments, we choose 5e-5 as the best learning rate. Furthermore, linear warm up schema is taken advantage of for the Bert model i.e. the pre-defined learning rate would be multiplied with a changing rate during the training.

Due to the restriction of BERT’s maximum sequence length, we limit the maximum sequence length for our batched data as 512 and batch size is 8.

5 Evaluation

Table 3 demonstrates the best model performance on validation dataset using different models. The best version refer to the version of the highest F1-micro score. All the other metrics are all evaluated on that version.

5.1 General observations

Metric analysis Precision@8 and precision@15 are micro-averaged precision scores, calculated by taking 8 and 15 labels with highest prediction probability and checking how many of them actually appear in the ground truths. The metric calculation method is from [7].

Model	F1-micro	F1-macro	Precision	Recall	Precision@8	Precision@15
Using High Level Diagnose Codes						
SLSTM	0.580	0.539	0.569	0.601	0.452	0.351
HAN	0.650	0.497	0.703	0.604	0.533	0.389
HSAN	0.655	0.505	**0.723**	0.539	0.522	0.382
CLSTM	0.670	**0.622**	0.659	0.681	0.530	0.381
BERT	**0.712**	0.534	0.701	**0.729**	0.538	0.363
Using Detailed Diagnose Codes						
SVM[5]	0.223	-	-	-	-	-
Logistic Regression[7]	0.242	-	-	-	-	-
HAN	0.246	0.068	0.165	0.424	0.288	0.221
BERT	0.251	0.078	0.173	0.453	0.267	0.210
SLSTM	0.253	0.112	0.087	0.451	0.158	0.146
HSAN	0.255	0.073	0.180	0.420	0.293	0.219
CLSTM	0.281	0.114	0.223	0.381	0.173	0.165
CBOW[5]	0.300	-	-	-	-	-
HA-GRU[5]	0.405	-	-	-	-	-
Bi-GRU[7]	0.417	-	-	-	-	-
CNN[7]	0.419	-	-	-	-	-
DR-CAML[7]	0.529	-	-	-	-	-
CAML[7]	0.539	-	-	-	-	-

Table 3: Model performance on MIMIC-III dataset

A general observation for all models is that, overall the recall is higher in comparison to precision, which reflects that the models tend to assign more labels than needed. This in turn reflects that the low level codes contains more noise for the prediction task, considering the limited amount of sampled data for experiment. This observation guided us to further carry out experiments on high level codes.

Compared with the prior approaches, our experiments show that a smaller discrepancy between F1-micro and F1-macro scores for high level code prediction. This can be interpreted as that the classes are more balanced for high level codes, as different diseases are pooled together into categories.

Model generalizability We observe an early overfitting issue on all models in training the sampled 20k MIMIC-III subset. One possible explanation is that the volume of training data is not sufficient enough for models to learn the general distribution. Another possibility is the brute truncation of the long input text. The maximum sequence length is set to 512 for BERT model and thus 91.81% of the input texts are truncated. For other models, 99.95% is truncated due to training speed limitation and GPU memory restriction.

Convergence rate Although the final ending point of the metrics are not as good as expected, fast convergence is observed across different architectures. Number of epochs needed for HAN, LSTMs and BERT models are approximately 2, 5, 10. This reflects the advantage of the newly proposed architectures. Although shallow, they have the same capacity as deeper models; and because they are shallow, the training time is significantly reduced.

Comparison with prior approaches [5] and [7]'s work generally achieved higher micro-F1 scores as shown in Table 3 (detailed code prediction part). However, their prediction model is limited to discharge summaries, while we do NOT filter any medical records which contains other categories such as nursing, radiology, physician, etc. The discharge summary is relatively better-structured (easier to extract sections based on hard-coded rules) and contains fewer than 1,900 sequences for an entry.

5.2 LSTM

Comparatively, CLSTM has consistent superior performance than SLSTM, we suppose that it takes advantage of disease title vocabulary appearing in the doctor note to "pin" the labels on those mentioned diseases. For lower level ICD-9 codes, the models are typically trained for 50 epochs and would start overfitting within 10 epochs; with higher level codes, the models overfit faster at 5 epochs. This is expected as the task is considerably easier for higher level codes. Though shallow, the models are able to overfit on train and even development set in 150 epochs, which reflects the sufficiency in model capacity and the intrinsic difficulty of the task.

Weighing mechanisms After adding `pos_weight` to the loss function `BCEWithLogitsLoss` aiming to balance the positive examples, it takes longer to converge. An additional weighing mechanism is used as per record positive weight, aiming to focus learning on records that have smaller number of labels to increase signal-to-noise ratio.

5.3 HAN

The performances of HAN and HSAN are similar, with F1-scores at around 0.65. They both converge rather quickly. As early as the 2nd epoch, both models can already produce reasonable outputs. Further training does not seem to make much improvement but only causes over-fitting on the training set.

Self-attention is said to have the advantages of modeling interactions between input words and minimizing total computational complexity per layer [13]. For this task, it fails to significantly improve the performance metrics. But it does almost half the training time, lowering it from 25.09s per iteration to 12.49s/iter.

5.4 BERT

The deep architecture consisting of 128 layers endows the BERT model the ability of fitting given data. BERT achieves the highest micro-F1 score among all models we experimented. Figure 2(a) and Figure 2(b) indicates the model is able to make perfect predictions on training dataset. However, for the never-seen validation data, it can only achieve up to 0.712 micro-F1 score. Furthermore, the loss on validation data increases after 20 epochs, indicating an overfit issue.

It is observed from Table 3 that BERT cannot guarantee best performance in all all metrics. e.g. Its macro-F1 score is lower than CLSTM's, showing unbalanced F1 scores on different classes. The gap between micro-F1 and macro-F1 is larger using detailed diagnose codes as ground truth.

Team Member Contributions

Ziyu Qiu Explored ICD structure and experimented with different data preprocessing mechanisms.

Yezhou Ma Explored related works and pre-trained models, experiment around BERT model.

Ta-Wei Mao Responsible for data preprocessing.

Jingxuan Sun Helped understanding data format, debugging data and trained two LSTM networks

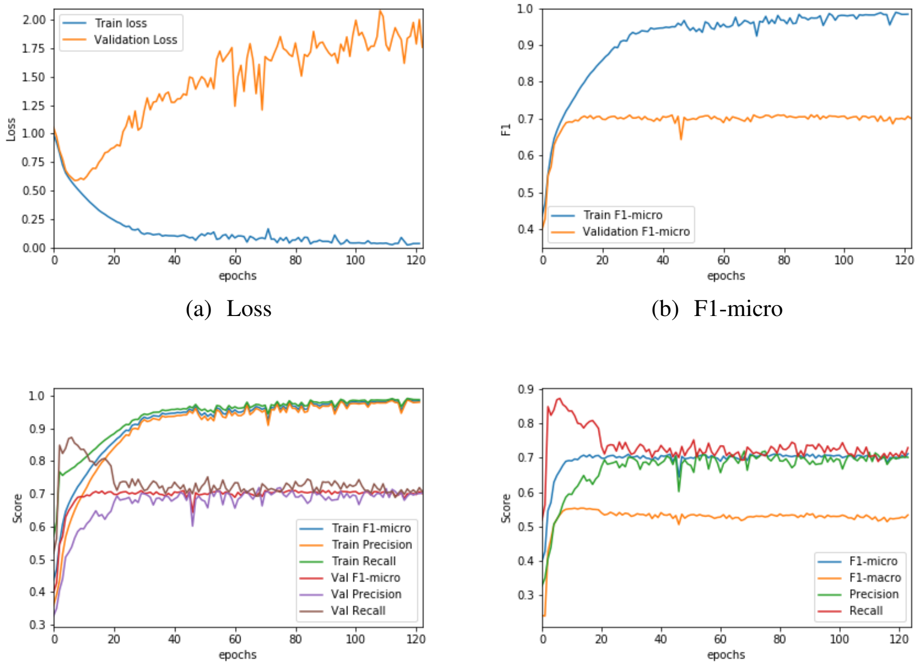
Yixue Wang HAN and HSAN models.

References

[1] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[2] Emily Alsentzer, John R Murphy, Willie Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, and Matthew McDermott. Publicly available clinical bert embeddings. *arXiv preprint arXiv:1904.03323*, 2019.

[3] Saadullah Amin, Günter Neumann, Katherine Dunfield, Anna Vechkaeva, Kathryn Annette Chapman, and Morgan Kelly Wixted. Mlt-dfki at clef ehealth 2019: Multi-label classification of icd-10 codes with bert. *CLEF (Working Notes)*, 2019.



(a) Loss

(b) F1-micro

(c) Comparison between train and val

(d) Metrics on Validation set

Figure 2: Model performance of fine tuning on Clinical BERT using high level diagnose codes

[4] 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.

[5] Tal Baumel, Jumana Nassour-Kassis, Raphael Cohen, Michael Elhadad, and Noemie Elhadad. Multi-label classification of patient notes: case study on icd code assignment. In *Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[6] Francisco Duarte, Bruno Martins, Cátia Sousa Pinto, and Mário J Silva. Deep neural models for icd-10 coding of death certificates and autopsy reports in free-text. *Journal of biomedical informatics*, 80:64–77, 2018.

[7] James Mullenbach, Sarah Wiegreffe, Jon Duke, Jimeng Sun, and Jacob Eisenstein. Explainable prediction of medical codes from clinical text. *arXiv preprint arXiv:1802.05695*, 2018.

[8] Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3:160035, 2016.

[9] Donna J Cartwright. Icd-9-cm to icd-10-cm codes: what? why? how?, 2013.

[10] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[11] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. *arXiv preprint arXiv:1904.03323*, 2016.

[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. *Bioinformatics*, 36(4):1234–1240, 2020.

[13] Qipeng Guo, Xipeng Qiu, Pengfei Liu, Xiangyang Xue, and Zheng Zhang. Multi-scale self-attention for text classification, 2019.