Harnessing the Natural Language Processing and Deep Learning in Clinical Research Informatics: a Case Study of MIMIC Dataset

Kaizhen Zhang, Master[1], Yingying Chen, Master[1]

[1]Georgia Institute of Technology, Atlanta, Georgia

Abstract

Predicting ICD-9 codes from the clinical notes is challenging but valuable for clinical practices. Emerging deep learning has been exploited in natural language processing (NLP) and facilitated the data mining of the clinical notes. Although various NLP systems have been developed for processing clinical texts, there is considerable room for future improvement in both word representation and model architecture in order to increase the prediction accuracy. In this project, two approaches of extrinsic evaluation for NLP based neural networks, top-level 18 categorical ICD-9 codes and top20 generic ICD-9 codes, were implemented to study the clinical notes from MIMIC III dataset. For this multi-class multi-label problem, the model of 2-layer LSTM networks with an embedding layer were trained with different word sequence length. Here the conventional neural network (CNN) was selected as the baseline model for comparison. The performance metrics, i.e., precision, recall and F1 score, were obtained for both the entire codes and each category. The results indicated that the model with word sequence of 1500 have the best performance, either too long or too short word length would result in lower accuracy. The model trained with the word sequence length of 1500 returned the entire system prediction precision of 77.74%, 75.60% for test datasets in "top-level 18 categories" and "top20".

I. Introduction

The assignment of ICD-9 codes is a critical step for clinical practices. This may manifest as critically important for clinical settings in the intensive care units (ICUs) due to the peculiar surveillance and superior medical cares provided[1]. Traditionally, this was evaluated by physicians on the basis of clinical knowledge and experiences or manually assigned by the medical coders. With the advance of clinical research informatics, myriads of statistical analysis and machine learning algorithms have been exploited with the promise of automatically tagging ICD-9 codes. However, the precision of these model-based approaches are not yet sufficient for clinical decision making[2]. One reason is the coarse medical profiles of patients used in these studies.

To date, electronic health records (EHRs), a structured data composed of patients' health status, are widely adopted as the features that influence and determine the future outcome. However, a significant amount of information including the doctors' impression of any performed tests and/or recommend treatments if necessary could not be presented in EHRs. The inherent formation of EHRs limits the information to be attached. One promising solution is to extract and investigate information from relatively rich and narrative clinical notes. Clinical notes represent the continuous flow of interaction between the doctor and patient, which are ready to be exploited for diagnosis guidance and medical care systems improvement.

The last two decades has witnessed the onset of natural language processing (NLP) coupled with deep learning in healthcare applications with the goal of driving clinical decision support[3], identification of events or diseases of interest[4, 5] and assignment of medical codes[6-8]. Various NLP systems have been developed for processing clinical texts. One of the earliest NLP systems was developed in The Linguistic String Project[9], followed by the clinical NLP system MedLEE (Medical Language Extraction and Encoding System)[10]. Owing to the advance of machine learning algorithms, modern NLP systems center on learning-based methods. Such a representative includes learning embedding word2vec[11]. Recently, Kang et al [12] presented an open-source information extraction system Eligibility Criteria Information Extraction (EliIE) for and reported that the EliIE outperforms existing systems. Deep learning is suitable for completing clinical tasks, even dealing with dataset without explicit features or rules. RNNs were employed in predicting clinical events[13]. LSTM RNNs could be used to address the long term dependency and recognize patterns in multivariate time series of clinical measurements[14].

In this study, the main goal is to implement the beachmark RNN (recurrent neural network) preceded by feature extraction using NLP to assign the ICD-9 codes. Moreover, a baseline model based on CNN (conventional neural network) was established as a fair comparison. The parameters including interpretation of ICD-9 codes, word

embeddings and neural network layers were tuned to establish the resilient model architecture and evaluate the effectiveness of the state-of-art deep learning models.

II. Methods

Data Preprocessing In this project, the clinical notes and ICD-9 Diagnoses of 38,597 distinct adult patients from 2001 to 2012 were retrieved from the publicly available multi-parameter intelligent monitoring in intensive care database MIMIC-III (Medical Information Mart for Intensive Care III) [15]. The notes that have the "Discharge summary" under the "category" column were selected to capture the complete information for a patient per admission. In terms of the instances of same admission ID and same "charttime", only the note that corresponding to the highest row ID was used for ease of table joint. For processing ICD-9 codes, we only consider the top level codes in the hierarchical structure, i.e., the first three digits of the codes. The codes were furthers processed by two approaches. To be inclusive, the top-level codes were summarized into 18 categories according to the "list of ICD-9 codes" based on the criteria released by International Statistical Classification of Diseases and Related Health Problems (referred to as **categorical ICD-9 codes**). Second, the top20 top-level ICD-9 codes were chosen (referred to as **top20 generic ICD-9 codes**). Figure 1A and B showed the counts of codes by using these two approaches. From Fig. 1A, it was suggested that the ICD-9 distribution was imbalanced. Particularly, categorical 7 is dominant while categorical 11 occurs rarely. Fig. 1B showed the top 20 generic ICD-9 codes. They covered at least 85% of the dataset (data not shown) and thus were considered to be a reasonable threshold. Then the cleaned notes table was joined with categorical ICD-9 codes and top20 generic ICD-9 codes, respectively. Therefore, only admissions with both notes and ICD-9 codes were kept. Fig 1C denotes the statics of note length for such two datasets. From Fig 1C, it was concluded that the two processing methods didn't contribute to the significant change to the length of notes of interest (p value>0.05). To train and evaluate the model, the whole dataset was randomly separated into the train dataset (50%), evaluation dataset (25%) and test dataset (25%).

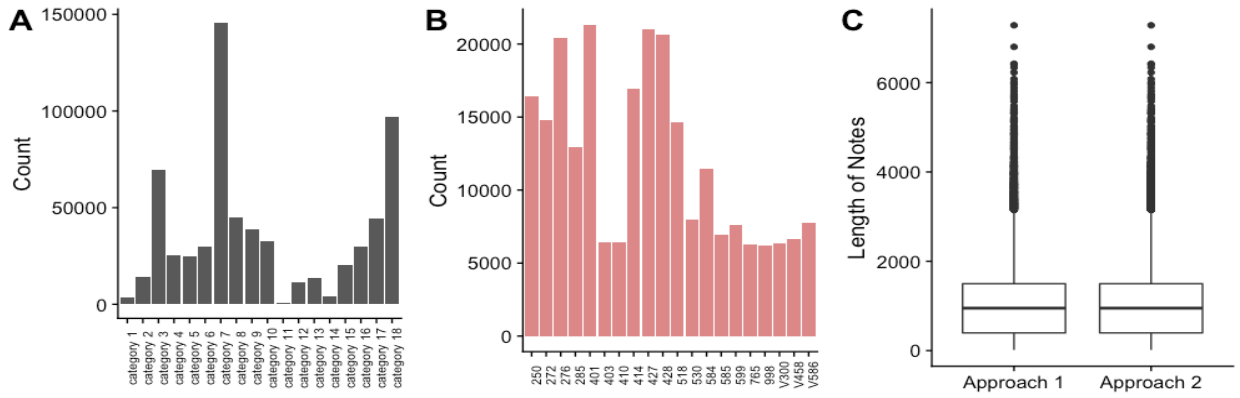

Fig. 1 (A) Counts of categorical ICD-9 codes (B) Counts of top20 generic ICD-9 codes (C) Boxplot of length of notes by two approaches

Feature Extraction In this project, the preprocessed note data was further manipulated to filter out the unrelated information in '[]' and '<>' inclusively, convert all the letters to lowercase and eliminate all non-alphabetic characters with regularization. 99885 unique words was then extracted from the resultant clinical notes and tokenized. Subsequently, the tokenization was used to generate word sequence for each note. Although different clinical note contains different amount of words, the word sequence was trimmed or padded to the same length. To interpret the impact of word sequence length on the model accuracy, different length, such as 50, 500, 1000 and 1500, 2000 were used in this project, since the median word length for text note is 1031. Furthermore, the key point here is to show how many words were needed to tag a text note correctly, instead of covering all the text notes, which would be cumbersome. Before training the LSTM model, the word sequence was converted to vectors via an embedding layer, formed with selected pre-trained GloVe model [16]. The pre-trained GloVe model used here was from 840B tokens on common crawl, since it contains 2.2 million vocabularies and most of the words in the clinical notes. Furthermore, the word2vector trained by ourselves performs no better than GloVe model.

Model Establishment. LSTM and CNN models has been widely applied for NLP text classification. Although CNN advances for its high training efficiency, LSTM wins over its accuracy. To make a good comparison between these

models, here, a LSTM model with 2 LSTM layers and a CNN model with 2 convolutional layers are built and trained, respectively. To overcome the vanishing gradient problem in RNN, model with two LSTM layers were implemented. The architecture of the model contains 1 embedding layer and two LSTM layers, as shown in Table 1. Here, the 'trainable' in embedding layer was set to 'False' to train the model efficiently. Meanwhile, 'return_sequences' was set as 'True' for the 1st LSTM layer so that its hidden states would be taken as input for the stacked 2nd LSTM layer. The dropout layer and the batch normalization layer was to avoid the overfitting and improve the computational efficiency. Since each clinical note might be tagged with multiple labels, the 'activation' of dense layer was set as 'sigmod' and the 'loss' in the model compile was set as 'binary_crossentropy', as illustrated below:

$$loss = -\sum_{i=1}^{N} [y_i \log p_i + (1 - y_i) \log(1 - p_i)]$$

where N the number of categories; y_i binary indicator (0, 1) for category i; p_i predicted probability for category i.The 'optimizer' for the model compile was selected as 'rmsprop', which was recommended for RNN. In the model training, the early stopping criteria is based on the validation loss. If the validation loss hasn't improved in 5 epochs, the model training would be stopped automatically.

Table 1 Architecture of the LSTM model with two LSTM layers for the input sequence length of 500

Layer (type)	Output Shape
input_layer (InputLayer)	(None, 500)
embedding_layer (Embedding)	(None, 500, 300)
lstm_1 (LSTM)	(None, 500, 256)
dropout_1 (Dropout)	(None, 500, 256)
batch_normalization_1 (BatchNormalization)	(None, 500, 256)
lstm_2 (LSTM)	(None, 64)
dropout_2 (Dropout)	(None, 64)
batch_normalization_2 (BatchNormalization)	(None, 64)
dense_1 (Dense)	(None, 18)

Likewise, the CNN model architecture was shown in Table 2. The CNN model contains 1 embedding layer and two convolution1D layers. To increase the computing efficiency, MaxPooling layer is applied right after each convolutional layer. Furthermore, the dropout layer is applied to tackle with the overfitting issue.

Table 2 Architecture of the CNN model with two convolutional layers for the input sequence length of 500

Layer (type)	Output Shape
embedding_layer (Embedding)	(None, 500, 300)
zero_padding1d_1 (ZeroPaddingLayer)	(None, 502, 300)
conv1d_1 (Conv1D)	(None, 500, 256)
zero_padding1d_2 (ZeroPaddingLayer)	(None, 502, 256)
conv1d_2 (Conv1D)	(None, 500, 256)
max_pooling1d_1 (MaxPooling1D)	(None, 250, 256)
zero_padding1d_3 (ZeroPaddingLayer)	(None, 252, 256)
conv1d_3 (Conv1D)	(None, 250, 128)
zero_padding1d_4 (ZeroPaddingLayer)	(None, 252, 128)
conv1d_4 (Conv1D)	(None, 250, 128)
max_pooling1d_2 (MaxPooling1D)	(None, 125, 128)
flatten_1 (Flatten)	(None, 16000)
dense_1 (Dense)	(None, 1024)
dropout_1 (Dropout)	(None, 1024)
dense_2 (Dense)	(None, 128)
dropout_2 (Dropout)	(None, 128)
dense_3 (Dense)	(None, 18)

Model Evaluation To better evaluate the model, different performance matrices, i.e., precision, recall and F1-score were obtained for both the category and the entire multi-classification system. Specially, the following matrices were used[17]:

For category i:

Precision: $p_i = \frac{|Y_i \cap Z_i|}{|Y_i|}$ Recall: $r_i = \frac{|Y_i \cap Z_i|}{|Z_i|}$ F1 score: $f_i = \frac{2 * p_i * r_i}{p_i + r_i}$ Accuracy: $a_i = \frac{|Y_i \cap Z_i|}{|Y_i \cup Z_i|}$

For the entire multi-classification system:

Precision: $P = \sum_{i=1}^{N} w_i p_i$ Recall: $R = \sum_{i=1}^{N} w_i r_i$ F1 score: $F = \sum_{i=1}^{N} w_i f_i$ Accuracy: $A = \sum_{i=1}^{N} w_i a_i$

Where Y_i is the set of samples predicted as true for category i; Z_i is the set of samples labeled as ground true for category i; N is the number of categories; $|Y_i|$ is the number of sample in set Y_i; w_i is the percentage of amount of category i in the total amount of all categories.

Implementation The data cleaning and transformation were conducted in the Scala 2.2.0. Model was built taking advantage of the packages of keras, sklearn and ntlk in python 3.6. The model trainings were conducted locally on MacBook Pro with processor of 2.9 GHz Intel Core i7 or Amazon EC2 server with 4-GPU. Data visualization would be realized in gg-plot2 of R studio.

III. Results & Discussion

This section focuses on (i) the impact of word sequence length on the LSTM model, (ii) comparison between LSTM model and CNN model, (iii) two extrinsic evaluation approaches on LSTM model.

Impact of word sequence length on the LSTM model

The clinical notes contain various length of words and it was expected that the length of word sequence would have a profound impact on the performance of the LSTM model. If the selected length is too short, the meaningful information is likely to be truncated. However, it is not necessarily true that a longer word sequence would lead to better performance of deep learning models. To investigate the impact of the word length on the LSTM model, the LSTM model is trained with the word length of 50, 500, 1000, 1500 and 2000 for the training dataset of the top20 generic ICD-9 codes. The evaluation functions, e.g. accuracy, precision, recall, F1-score for the test datasets are recorded and analyzed, respectively, as shown in Fig. 2. Although the accuracy increases monotonously from 81.15% to 88.44%, along with extent of the word sequences, the other evaluation functions encounter peak values at the word length of 1500. Obviously, the evaluation functions, e.g. precision, recall and F1 score, are improved along with the word sequence length increasing from 50 to 1500, due to the richer information conveyed by longer word sequence. Precision climbs from 52.31% at the word length of 50 to 75.60% at the word length of 1500. Similarly, recall climbs from 23.18% to 56.10%, F1 score climbs from 28.65% to 62.67%. However, the precision and F1-score drops when the word sequence increases from 1500 to 2000, though the recall is increasing substantially. The reason is that padding the short text notes to the specified length introduces redundant information, raising the similarity of the text notes labeled with different ICD-9 codes. Furthermore, the computing time gains almost proportional to the word sequence length. Hence, trade-off between the model performance and computing time indicates that the reasonable word sequence length is 1500 for this dataset. Unless specified otherwise, the word sequence length is 1500 in the following sections.

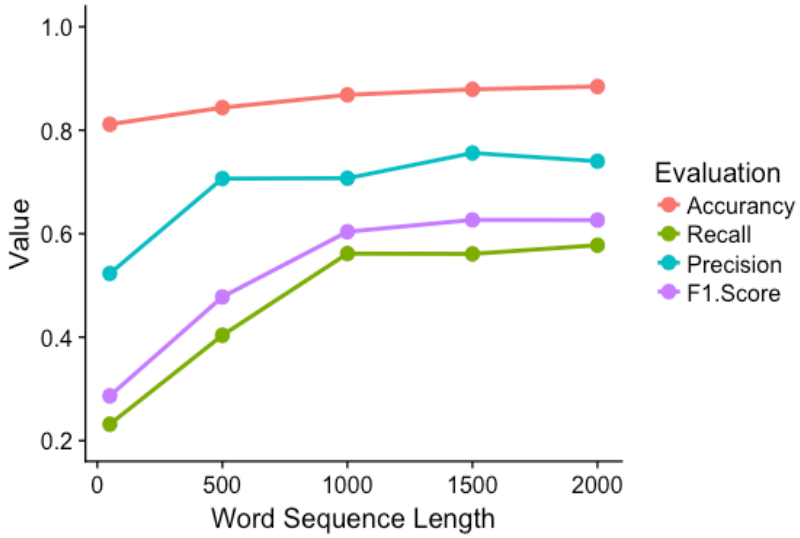

Fig. 2 Impact of word sequence length on the LSTM model trained for top20 generic ICD-9 codes

LSTM model versus CNN model

Currently, CNN and RNN are two widely used deep learning models. To leverage the deep learning in our case, Convlution1D CNN model was selected as the baseline model and compared with LSTM model. The LSTM model and CNN model are trained on the same dataset with the same computing resources. To make a fair comparison, we trained the CNN model with the word sequence length of 500, 1000 and 1500 for the training dataset of top20 generic ICD-9 codes. Then we use the testing dataset to evaluate the trained model and compare the evaluation results with those of LSTM model, as shown in Fig. 3. The results show that the LSTM model has over 15% higher precision, recall and F1 score for all the different word sequence lengths, although training CNN models could be completed in shorter time. For CNN, the model performs best at the word sequence length of 1000, about the median length of all the text notes, indicating that CNN is more sensitive to the redundant information introduced via padding. Hence, CNN would performance worse than LSTM to classify the clinical notes, whereof the length varies lots from different recorder. Although adding more convolutional layers should improve the CNN model, it cannot tackle the redundant information from padding in essence. Hence, our results suggested LSTM is more suitable to deal with the sequential words and would performance better than CNN intrinsically.

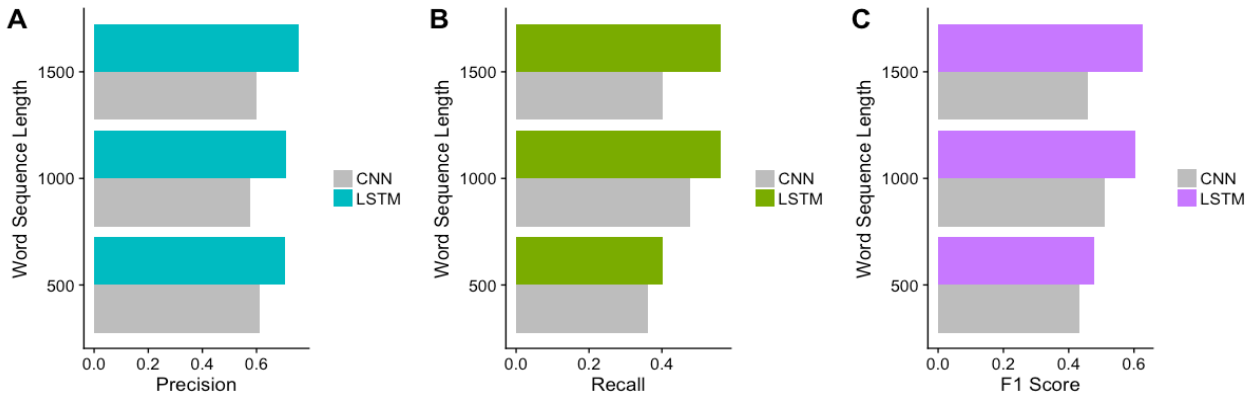

Fig. 3 Comparison between the LSTM model and CNN model

Two extrinsic evaluation approaches on LSTM model

LSTM model was then trained on two pre-processed datasets mentioned above: (i) top-level 18 categorical ICD-9 codes, (ii) top20 generic ICD-9 codes. Although various word sequence lengths have been used for training purpose, only the one with best performance is analyzed and discussed below. Here, the evaluation of the LSTM model focuses on not only the overall performance on the entire system but the individual category or generic ICD-9 code.

Top-level 18 categorical ICD-9 codes classification The LSTM model is trained for the dataset of top-level 18 categorical ICD-9 codes with the input word sequence length of 1500. For the overall performance on the entire multi-classification system, its precision, recall and F1 score are 77.74%, 60.90% and 63.95% (as shown in Fig.4), respectively, comparable with the results in literature[6]. The further analysis on the refined category reveals the main issue hindering the performance of the LSTM model, indicating the possible solution. The precision, recall and F1 score of the model on the test dataset is illustrated in Fig.4. Obviously, the model works at least fair for all categories except category 1, 11, 13, and 14. For category 1, 11, 13 and 14, the high precision shows that the predicted labels are trustworthy, while the low recall means that those categories is difficult to be distinguished from the other categories. Given the count of the notes for each category, as shown in Fig. 1(A), it is noticeable that all the categories with lower recall have less training dataset due to the rare disease represented by the ICD-9 code. This issue could be solved by collecting more data for the rare disease.

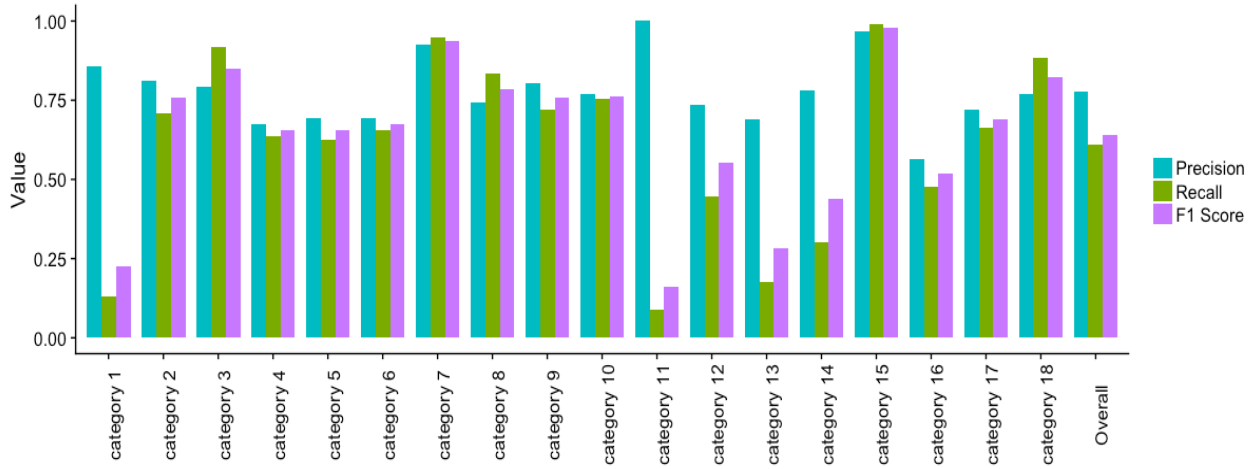

Fig. 4 Precision, recall and F1 score for each category and overall for top-level 18 categorical ICD-9 codes

Top20 generic ICD-9 codes classification Similar with the top-level 18 categorical ICD-9 codes classification, LSTM model is trained for the dataset of top20 generic ICD-9 codes with the word sequence length of 1500 here. To evaluate the model performance on the entire multi-classification system and each generic code, the precision, recall and F1 score are shown in Fig.5. For the overall performance, the precision, recall and F1 score are 74.00%, 57.79%

and 62.62%, respectively. All of these less notes under each generic code. Meanwhile, it works poorly on category 285, 998, V458, V586, due to two types of reasons. The first one is being categorized wrongly to the category under the same generic disease for lack of information; For example, V586, together with generic code 599 584, 585, belongs to the disease of the genitourinary system. Hence, it is likely that V586 are categorized wrongly as category 584, 585, 599. Similarly, V485 could be wrongly categorized as 518. This issue could be addressed by using distinguishable words for the generic notes under the same top-level category before data preprocessing. Another reason for this issue is the small data set for given categories, such as 285, 998. Therefore, it is important to collect large enough samples for a given disease and then implement deep learning model for prediction.

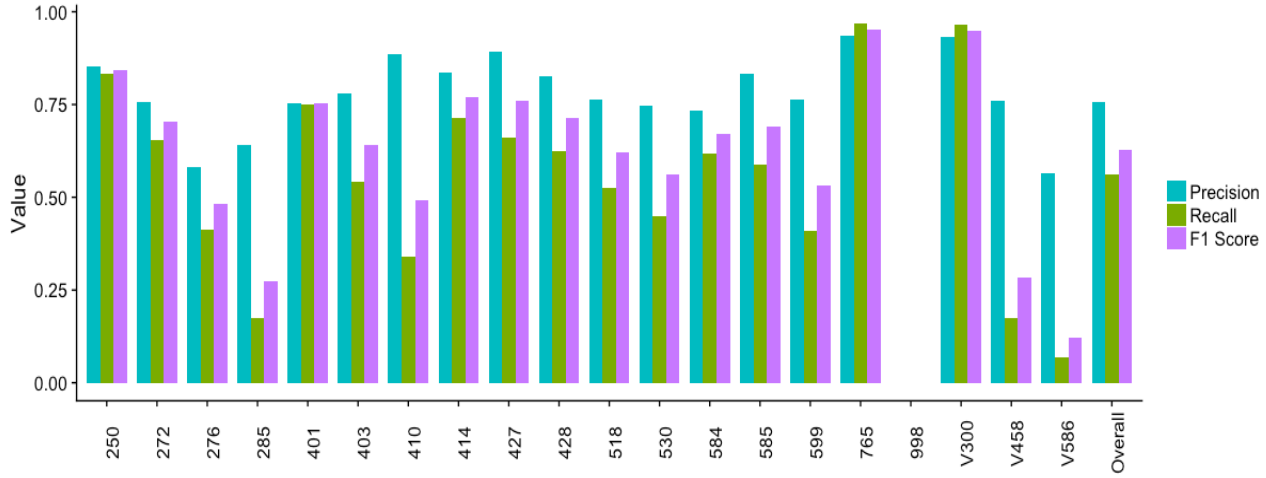

Fig.5 Precision, recall and F1 score for each category and overall for top 20 generic ICD-9 codes

IV. Conclusions

In this project, a multi-class multi-label LSTM network has been built to predict the ICD-9 codes given the clinical notes. As to the ICD-9 codes, we evaluate the prediction of the top-level 18 categorical ICD-9 code and top20 generic ICD-9 code as proof-of-concept inclusive and specific study. The impact of the word sequence length on the model performance reveals that proper length of word sequence is critical in model performance. Here, model trained with the word sequence length of 1500 performance best, returning the predicted overall precision as 77.74%, 75.60% for "top-level 18 categorical codes" and "top20 generic ICD-9 codes", respectively. The comparison between the CNN model and LSTM model illustrates that LSTM performs much better than CNN model for clinical notes classification. Moreover, analysis of the LSTM model performance on each category or generic codes indicates that more data for the rare diseases should be collected in order to further improve the model performance. Our study indicates the LSTM model could be a useful tool for multi-classification from plain texts after NLP treatment. Further work could be focused on obtaining pre-trained word vectors with the specialization of medication. Additionally, both the parameters of LSTM and CNN models could be tuned for optimization. This study offers a practice of using current deep learning models for the supervised learning of plain notes.

Supporting Information:

All the related codes and slides for presentation could be found in the link:

Datasets for model: https://drive.google.com/drive/u/1/folders/1PSjmBQcfvhVWgEAjAP8367UJ7cDds3VA

GitHub for code: https://github.com/CarsonZhang121/BDH_Project

Presentation: https://youtu.be/U1C_d0HCOGQ

Reference:

[1] R. Davoodi and M. H. Moradi, "Mortality Prediction in Intensive Care Units (ICUs) Using a Deep Rule-based Fuzzy Classifier," *Journal of biomedical informatics,* 2018.

[2] J. I. Salluh and M. Soares, "ICU severity of illness scores: APACHE, SAPS and MPM," *Current opinion in critical care,* vol. 20, no. 5, pp. 557-565, 2014.

[3] D. Demner-Fushman, W. W. Chapman, and C. J. McDonald, "What can natural language processing do for clinical decision support?," *Journal of biomedical informatics,* vol. 42, no. 5, pp. 760-772, 2009.

[4] N. Afzal *et al.*, "Natural language processing of clinical notes for identification of critical limb ischemia," *International Journal of Medical Informatics,* 2017.

[5] H. Ma and C. Weng, "Prediction of black box warning by mining patterns of Convergent Focus Shift in clinical trial study populations using linked public data," *Journal of biomedical informatics,* vol. 60, pp. 132-144, 2016.

[6] J. Huang, C. Osorio, and L. W. Sy, "An Empirical Evaluation of Deep Learning for ICD-9 Code Assignment using MIMIC-III Clinical Notes," *arXiv preprint arXiv:1802.02311,* 2018.

[7] D. T. Heinze and M. L. Morsch, "Automatically assigning medical codes using natural language processing," ed: Google Patents, 2005.

[8] E. Scheurwegs, K. Luyckx, L. Luyten, W. Daelemans, and T. Van den Bulcke, "Data integration of structured and unstructured sources for assigning clinical codes to patient stays," *Journal of the American Medical Informatics Association,* vol. 23, no. e1, pp. e11-e19, 2015.

[9] N. Sager, C. Friedman, E. Chi, C. Macleod, S. Chen, and S. Johnson, "The analysis and processing of clinical narrative," *Medinfo,* vol. 86, pp. 1101-1105, 1986.

[10] N. Sager, C. Friedman, and M. S. Lyman, *Medical language processing: computer management of narrative data*. Addison-Wesley Longman Publishing Co., Inc., 1987.

[11] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781,* 2013.

[12] T. Kang *et al.*, "EliIE: An open-source information extraction system for clinical trial eligibility criteria," *Journal of the American Medical Informatics Association,* vol. 24, no. 6, pp. 1062-1071, 2017.

[13] E. Choi, M. T. Bahadori, A. Schuetz, W. F. Stewart, and J. Sun, "Doctor ai: Predicting clinical events via recurrent neural networks," in *Machine Learning for Healthcare Conference*, 2016, pp. 301-318.

[14] Z. C. Lipton, D. C. Kale, C. Elkan, and R. Wetzel, "Learning to diagnose with LSTM recurrent neural networks," *arXiv preprint arXiv:1511.03677,* 2015.

[15] A. E. Johnson *et al.*, "MIMIC-III, a freely accessible critical care database," *Scientific data,* vol. 3, p. 160035, 2016.

[16] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532-1543.

[17] D. M. Powers, "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation," 2011.