NATURAL LANGUAGE PROCESSING OF CLINICAL NOTES FOR IDENTIFYING DIAGNOSIS AND PROCEDURES USING NEURAL NETWORKS

Siddhartha Nuthakki

School of Informatics and Computing-IUPUI,

Under the Guidance of

Dr.Saptarshi Purkayastha Ph.D.,

Assistant Professor, Health Informatics,

Indiana University-Purdue University, Indianapolis

Fall-2018

Accepted by the Faculty of Indiana University, in partial fulfillment of the requirements for the degree of Master of Science in Health Informatics

Mater's project committee

_____

Dr.Saptarshi Purkayastha , Health Informatics,

Assistant. Professor

_____

Dr. Judy Gichoya, MD MS, Oregon Health &

Science University

Table of Contents

LIST OF TABLES

LIST OF FIGURES

Acknowledgments

I would like to thank my professor Dr.Saptarshi Purkayastha who have helped me a lot and provided me with best of his knowledge to make this project a success. I would like to thank my friends and family for the support.

Abstract

Coding diagnosis and procedures in medical records is a critical process in the healthcare industry. It is important at various levels, from creating accurate billing, getting reimbursement from payers and creating standardized patient care record. In the US, Billing and Insurance-related (BIR) activities costed around $471 billion in 2012 (Jiwani, Himmelstein, Woolhandler, & Kahn, 2014) which accounts for 25% of all U.S. Hospitals spending. Additionally, coding is a tedious, tiresome process requiring patience and concentration to minute detail. The objective of the study is to build a natural language processing model which can map clinical notes to medical codes and predict the final diagnosis from unstructured entries of history of present illness, symptoms at the time of admission etc. Previous studies have demonstrated that deep learning models perform better at this task than conventional machine learning models. We employed the state-of-the-art deep learning method ASG Weight-Dropped Long Short-Term Memory (AWD-LSTM) on the largest emergency department clinical notes dataset MIMIC III with 1.2M notes for top-50. End-to-end machine learning methods were employed in our evaluations in contrary to manually defined rules. We used maximum vocabulary of 60,000 words and with a minimum frequency of 3. Our models predicted the top-10 and top-50 ICD-9 codes of diagnosis with 80.3% and 70.7% accuracy respectively. The second model could predict top-10 and top-50 ICD-9 codes of procedures with 80.5% and 63.9% accuracy respectively. Prediction of final diagnosis from either history of present illness or symptoms can let physicians identify promising treatment, which can potentially transform the conventional care of diagnosis followed by treatment. With the scores from the present models, the next step is to deploy on a small-scale real-world scenario and compare with human coders as gold standard. We believe further research of this approach can create highly accurate predictions which can ease workflow in clinical setting.

1. Introduction

Electronic health records (EHR's) contains information about a patient like patient past medical and medication history, symptoms, chief complaints, treatment, procedures and tests, final diagnosis, discharge medications and care notes or referral notes which can be tracked over time and is the best source for evidence based care among healthcare professionals. The usage of electronic health records (EHR's) has been skyrocketed for the past 10 years which is due to the implementation of 2009 Health Information Technology for Economic and Clinical Health (HITECH) Act. HITECH act provided $30 billion incentives for physicians and hospitals to adopt electronic health records and transform healthcare. A recent survey by the health information technology office of national coordinator (ONC), about 84% of the hospitals started using EHR which is assessed to be a 9-fold increase when compared to the adoption in year 2008. Moreover, adoption of basic and certified EHR's by office based physicians increased from 42% to 87% which more than doubled (Gehrmann et al., 2018).

Although EHR's are primarily designed to improve healthcare delivery and efficiency from an organization and institutional perspective, they have become rich source of data for clinical informatics professionals. There are numerous data sources which are de-identified and openly available for researchers. Few available resource is Medical Information Mart for Intensive Care (MIMIC) database are informatics for integration biology and the bedside (i2b2) database (Geraci et al., 2017). These datasets provide vast amount of information which are already being mined and explored in numerous ways. The data primarily exists in two forms: structured and unstructured data. International classification of diseases (ICD) codes, diagnostics and laboratory results, medications come under structured data. Whereas unstructured data include clinician progress notes and discharge summary. The extraction of information between structured and unstructured data varies a lot. Extraction of data from structured text is relatively easier than unstructured text. Statistical tests and machine learning

techniques can be applied relatively easier when compared to unstructured text. However, when compared to the structured portion of the data in the EHR like that of which is used for billing and administrative purposes, unstructured data like clinical notes is more nuanced but is primarily used for providers for thorough documentation of the care given. Typically, several admission notes, clinical notes, transfer notes and discharge summaries are associated with each patient history (Hung, Lin, & Lee, 2018). As this is unstructured data extracting information from this is very difficult and historically it requires wide range of manual feature engineering and mapping to ontologies which is why the adoption of such techniques are limited. One recent advancement in such techniques is usage of Natural Language Processing (NLP) which can extract only desired data form unstructured text, in association with analyzing structured data which can lead to even more understanding of the patient diseases and their prognosis (H. Liang, Sun, Sun, & Gao, 2017).

NLP can be used to extract data from unstructured text. In a clinical setting, NLP can be used to convert data from provider notes such as clinical notes, transfer notes, discharge notes into structured text in a predefined format which can be used for analysis. NLP has numerous advantages in this era of health information technology (Miotto, Li, Kidd, & Dudley, 2016). It is an invaluable tool for Health Information Management (HIM) professionals through which they directly process text thereby generating text which organization can use to enhance communication among care givers, enhance the cost effectiveness of clinical text documentation and processing, and also automate coding which is one of the important administrative task in documenting healthcare, billing and getting reimbursement (Liu et al., 2017).

However, NLP poses potential problems when dealing text that is ungrammatical, which consists of highlighted points like ‘bullet points’, telegraphic phrases and lack complete sentences. Heavy use of abbreviations and acronyms make clinical notes more ambiguous.

Clinical notes often contain words which has multiple meanings. For example, a word 'discharge' means getting discharged from the hospital or either bodily excretion; the word 'MD' can be referred to 'Doctor of Medicine' or 'mental disorder'. Researchers from Massachusetts Institute of Technology (MIT) set a new technique to over this which is called 'Topic modelling' (Wu, Jiang, Xu, Zhi, & Xu, 2017). It promises to significantly reduce the efforts of human in developing more accurate systems. It automatically identifies the topics in the documents by inferring relationships between important featured words. Topic modelling requires minimal human supervision, automating the algorithm to refine and revise the features (Shickel, Tighe, Bihorac, & Rashidi, 2018).

However, in the recent years, deep learning, which has proven advantages in numerous feature engineering techniques, has been employed widely in various fields, like speech recognition, image processing, NLP where it has proven potential. While coming to Natural Language Processing (NLP) it is employed to resolve many potential problems like machine translation, relationship extraction, named entity recognition, syntax parsing, word sense disambiguation, sentiment classification etc (Artetxe, Beristain, & Grana, 2018). with the availability of open source EHR's, deep learning in health care analytics has been a valuable resource for researchers. Few examples of recent advancements in the application of deep learning to EHR's is extracting relevant information like diagnosis to code for medical billing, extracting symptoms, procedures relevant to the study of interest from clinical notes.

As per Benjamin et al., (2018), the usage of deep learning in clinical context can be summarized as follows:

I. EHR information extraction.
    a. Single concept extraction
    b. Temporal even extraction
    c. Relation extraction and

d. Abbreviation expansion.

II. EHR representation learning

a. Concept representation

b. Patient representation

III. Outcome prediction

a. Static outcome prediction

b. Temporal outcome prediction

IV. Computational phenotyping

a. New phenotype discovery.

b. Improving existing definitions

V. Clinical data de-identification

These are the various topics on which deep learning can be applied on a clinical text. One task which we are closely associated is single concept extraction. Single concept extraction is a basic task extracting fundamental information like diagnosis, treatments and procedures from clinical text (Li, Liu, & Yu, 2018). Traditionally, several studies applied various natural language processing techniques to get varied level of success. However, with the complexity of the clinical notes, there is large room of improvements (Yang et al., 2018).

A study from Jagannatha et al. dealt with concept extraction problem like a sequence labelling task where the goal is to allocate one from the clinically relevant nine tags to each word from clinical notes. The tags that were allocated are medications, diseases within which there are be numerous sub tags like name of the drug, dosage of the drug, route of administration of the drug, indication of the drug, side effect of the drug, adverse drug reactions and disease severity. They performed analysis using several deep architectures like recurrent neural networks (RNN's) including Long Short Term Memory (LSTM) and Gated Recurrent Units (GRU's), bi directional LSTM's and various groupings of LSTM's including conditional

random fields (CRF's). They compared the baseline CRFs which was gold-standard for state-of-the-art technique for clinical concepts extraction (Zhu, Li, Conesa, & Pereira, 2018). They proved that RNNs outperformed CRF baseline accuracy with enhanced accuracy. This information is highly important for billing and coding. Various other studies discussed the application of deep learning to clinical concept extraction like recognition of named entities in clinical text by Wu et al., where pre-trained word embedding's of clinical text in Chinese using Convolutional Neural Networks (CNN) outperformed the CRF baseline accuracies (Bibault, Giraud, & Burgun, 2016).

In this study, we comprehensively investigate entity extraction from clinical text using deep learning. ASG Weight-Dropped Long Short-Term Memory (AWD-LSTM), a variant of one type of deep learning method (i.e., recurrent neural network), is deployed to recognize diagnosis and procedures from clinical text (Choi, Schuetz, Stewart, & Sun, 2017).

2. Background

Machine learning is one of the recent advancements in artificial intelligence. Machine learning has wide spread utilization like object identification in images, converting speech to text, matching relevant news items and products based on user interest and select relevant topics based on search results. These applications are currently being trained to use deep learning which has been the current area of interest for researchers. Traditional machine learning methods are limited in its application to natural data which is in unprocessed form (Huang, Dong, Duan, & Liu, 2018).

Initially building a machine learning system required lot of manual engineering with careful consideration and huge domain knowledge who can design an extractor which can transform raw data into usable format which the machine learning system can use. It seldom involves the creation of the classes suitable for a classifier system (Z. Liang et al., 2018). A method in

machine learning called ‘Representation learning’ involves the identification of representation necessary for the classification or detection when given raw data to the machine. Deep learning uses representation learning with varied levels of representation. It is obtained by composing multiple non-linear modules which can transform the representation at one level into next level which includes numerous complex transformations (Miotto et al., 2016).

Extracting features from the text is the basis of processing text documents in large number. Text features are the basic unit of processing text documents. Selecting important set of features by reducing the dimensions of work space is called feature extraction. During this process, irrelevant and uncorrelated features are deleted. Feature extraction has been showed to improve the accuracy of the model along with reducing time to train the model. Selecting features from the text document is called text feature extraction. Widely used methods for text feature extraction include mapping, fusion, filtration and clustering methods. Traditionally, feature extraction was done using handcrafted methods. Manually designing an effective and efficient feature is a time-consuming process. In contrary to that, deep learning offers an effective way to acquire features quickly from training data (Wu et al., 2017).

The advantages of deep learning over traditional manual engineering methods are that they are not designed manually; they are identified from data by the machine on its own. Deep learning methods require very little manual engineering, so large amount of data can be processed very easily by deep learning methods (Bibault et al., 2016). Deep learning can deal with unstructured data like images, sound, media, text and video very effectively than most of the traditional methods. Deep learning has shown promising results for numerous tasks of natural language processing like sentiment classification, topic modelling, language translation and question answering. The deep architecture allows solving even more complicated AI tasks. Few methods that are used in deep learning for feature representation are autoencoder,

convolutional neural networks, recurrent neural networks, deep belief network and restricted Boltzmann model etc.

Numerous researchers have developed several computational models which can be applied to general clinical natural language processing systems. Few examples are MetaMap4, KnowledgeMap6, MedLEE5 and rule-based methods which depends on existing medical vocabularies or medical dictionaries for named entity recognition. Few challenges are organized by clinical natural language processing community members to assess the performance of state-of-the-art models. During their challenges, most of the supervised learning methods with hand labelled features are the top performers. To even further enhance the accuracy of the models, researchers explored various feature engineering techniques within the available infrastructure of traditional machine learning techniques which includes ensemble learning methods that group multiple machine learning techniques, hybrid systems which group machine learning with unsupervised features generated during clustering algorithms (Reddy & Delen, 2018).

Traditionally named entity recognition is used to extract useful concepts like medications, diseases, lab tests from clinical text which can support clinical research. Conventional named entity recognition is named as sequence labelling task which aims to provide best label for the given text. Many machine learning techniques were used by researchers like maximum entropy (ME), conditional random fields (CRF's) and support vector machines (SVM's) (Lyu, Chen, Ren, & Ji, 2017; Miotto et al., 2016). Most of the traditional models used CRF's which is one of the effective method in conventional machine learning techniques (Huang et al., 2018). A typical clinical NER with state-of-the-art technique utilizes various linguistic levels (like prefix, suffix, and capitalization of letters), parts of speech tagging, word n-grams and concept identifier (semantic information). Few hybrid models use concepts like MetaMap, cTAKES.

More recent advancements are the use of multiple ensemble methods which combine multiple machine learning models (Beaulieu-Jones, Orzechowski, & Moore, 2018).

3. Literature review

Deep EHR: A Survey of Recent Advances in Deep Learning Techniques for Electronic Health Record (EHR) Analysis

In this paper a survey was conducted regarding the application of deep learning in learning a variety of tasks based on the available clinical data. Deep learning has been implemented in building up variety of applications for clinical use, extracting the required information, predicting the outcomes, de-identifying the patient data, analyzing the genomic and phenotypic data. This paper also discussed the limitations faced during implementing deep learning due to lack of proper infrastructure such as lack of benchmarks that is universally accepted, Interpreting the right models that should be applied for building models that are clinically useful, Different forms of raw data that is not even pre-processed. They have concluded by mentioning the pathways for future research in deep learning. The implementation EHR has drastically increased due to HITECH act. There is a nine fold increase in implanting EHR from 2018. They consists of entire demographic and clinical information of the patients. In the last few years dependencies of the features are identified by various statistical and traditional machine learning methods. In the search strategy articles were collected until 2017 with various key terms related to Electronic health records and Deep learning (Korvigo, Holmatov, Zaikovskii, & Skoblov, 2018). Since EHR implementation in increasing over years they are also using standard formats to record the information such as using ICD codes for diagnosis, LOINC for billing, CPT codes for procedures etc., a variety of machine learn models getting implemented to know the underlying issues and to use the available data meaningfully to improve the clinical outcomes of the patient. They have

mentioned hierarchical view of various deep learning techniques such as Multilayer perception, Convolutional neural networks, recurrent neural networks, RBM, and Auto encoders. They have also mentioned about many engineering applications and in building the auto mapping ontologies to extract the relevant information from clinical notes such as extracting the medical concepts such as procedures, diseases and treatments. Events extraction by mapping the concepts with the time, extracting the relationship between the concepts and unstructured notes and diagnosis, expanding the abbreviations in the clinical notes. They have also mentioned about understanding natural relationships and forming clusters to extract the relevant information. Finally, they have concluded that the human interpretability is elusive task tough various deep learning models have been implemented. Since they are used in clinical decision making the practitioners are not ready to trust such methodologies. They have also mentioned that applying deep learning in medical records will be ever going research in future (Artetxe et al., 2018).

Applying deep neural networks to unstructured text notes in electronic medical records for phenotyping youth depression (Reddy & Delen, 2018).

In this study, they have used deep learning to identify the phenotypic inclusion criteria for a study related psychiatric diagnosis in youth. This study was conducted on documented 861 medical records. There is about 60% errors in identifying the participants or populations in clinical settings. Mapping the Electronic medical records has become as source to identify the cohorts for the study from large data bases. Many procedures such as Natural language processing (NLP), Machine learning and learning are used to identify and extract the useful information form unstructured data. Now a days NLP is being used for variety of tasks such as identifying the phenotypes of the patients, Boosting the capability of extracting data from charts, Identifying the Adverse drug effects, and identifying the risks to prone a disease in

children's and adult's. Latest methodologies include the identification of depression and bipolar patients based on the tweets in twitter (Li et al., 2012). By combining NLP and Machine Learning technologies have advanced in finding the diagnosis of the patients form text available in discharge summaries by extracting medical texts by reasoning and mapping. Coming to the methodology they have deidentified the patient's records by using the Perl-based software. Their main aim is to identify the patients between the age groups 12-18 having Major disorders of depression. Main other categories were excluded from the study such as epilepsy, brain injuries, autism etc., they have used methods the used along with NLP packages such as A brute force search method and applying Neural networks. The brute force search method works by identifying such key terms in the free text available and neural networks works by predicting the relationships between the various non-linear inferences. They found that sensitivity and specificity obtained by this two models is low. So, they have combined these two method s by building a model DL+0 which yielded better results such as sensitivity of 93.5% and specificity of 68% with a precision of 77%. They have concluded that this model will be further more developed by implementing this model in large datasets. They also mentioned that further research is useful in building an alternate to neural networks and brute force methods (Shickel et al., 2018).

4. Methodology

Figure 1 shows an overview of the methodology pipeline. Our methodology consists of the following steps: data preprocessing, building language model and classifier model. Specifically, we use Python for data preprocessing: Python, numpy, pandas, sklearn for feature extraction; fastai, pandas_ml, scipy for training and testing. We use Google virtual machines (NVIDIA Tesla V 100 GPU) and school virtual machines (Quadro P6000 and GetForce GTX 1080) to run our experiments. Sections below explains methodology in detail.

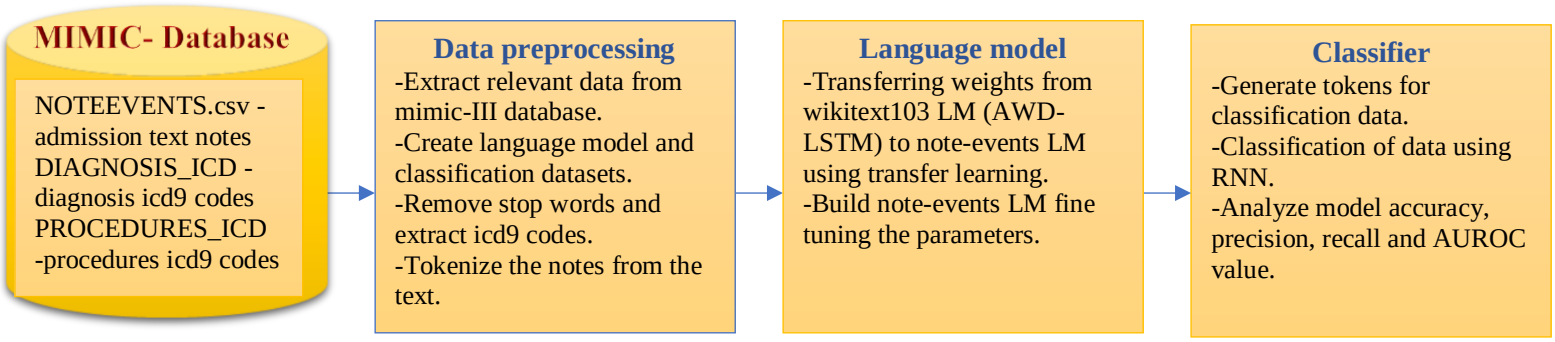


Figure 1: Methodology pipeline overview

4.1 **Data preprocessing**

MIMIC-III is a large database containing patient's data admitted to critical care units of a large tertiary care hospital. From that database note-events, diagnosis_icd, procedures_icd tables were selected which are relevant to the research question. Each dataset is having numerous columns from which necessary columns were extracted. From the note-events table the following columns were chosen for analysis: subjectID, admissionID and discharge notes (text). The note-events table contains 2.0M rows with text divided into subsections like admission date, discharge date, date of birth, sex, service, chief complaint, history of present illness, past medical history, medications on admission, allergies, family history, social history, physical exam at the time of admission, laboratory studies, brief summary of hospital course per disease like COPD, hypertension, hyperglycemia, hypothyroidism, cardiovascular, depression etc., discharge condition, discharge status, discharge medications, follow-up plans and final diagnosis.

Likewise, from diagnosis_icd table the following columns were chosen: subjectID, admissionID and icd9 code for the diagnosis. The diagnosis table contains 651K rows with 6984 unique diagnosis. For a combination of subjectID and admissionID there are numerous diagnosis ranging from 1 to 38 with a sequence number denoting the order with which the diagnosis is relevant to the patient (Choi et al., 2017).

Similarly, from procedures_icd table the following columns were chosen: subjectID, admissionID and icd9 code for procedures. The procedures table contains 240K rows with 2009 unique procedure codes (Xiao, Choi, & Sun, 2018). For a combination of subjectID and admissionID there are numerous procedures ranging from 1 to 14 with a sequence number denoting the order with which the procedure is relevant to the patient. Table 1 describes the size of the dataset and their respective unique values.

Category	**No of rows**	**Unique values**
Note-events	2083180	2023185
Diagnosis	651047	6984
Procedures	240095	2009

Table 1. MIMIC-III descriptive statistics

Preprocessing these three datasets produced two separate datasets. Initially, diagnosis and procedures datasets were merged using Python Dask based on columns subjectID and admissionID. Inner joining these two tables resulted in a table with 3.0M rows. This table consists of subjectID, admissionID, diagnosis_icd9 code and procedures_icd9 code. This table is merged with note-events table containing 2.0M rows which resulted in an unmanageable 800GB data divided into multiple csv files with each file size ranging from 8 to 12GB. A file with 10GB size was chosen from this large number of files and this dataset is divided into train and test of 90:10. Training the language model using this file depicted 51hours for 1 epoch using GetForce GTX 1080 provided by our school. Realizing the potential problem with the size of the data, further filtering of the diagnosis and procedures tables were performed (Bibault et al., 2018).

Having known that each subject with an admissionID is having multiple diagnosis and procedures, we restricted the data by considering only top one diagnosis and procedure for each admission based on the sequence number. It restricted the dataset size of diagnosis and

procedures to 58929 and 52243 rows respectively (Beaulieu-Jones et al., 2018). Table 2 explains the size of each dataset with their respective unique values. Unlike that of the previous merging of datasets (diagnosis merged with procedures followed by merging with note-events), these two datasets were individually merged with note-events data. Inner joining the diagnosis and procedures table with note-events dataset resulted in 1.8M and 1.7M rows respectively. As both the datasets are similar in size, diagnosis dataset was divided into train and test of 90:10 ratio for training language model. The time for training 1 epoch depicted 26hours on GetForce GTX 1080 which would take 800 to 900 hours to train a language model and classifier for 10 epochs. As allocating the resources for this amount of time is not feasible, the dataset is further filtered.

Category	**No of rows**	**Unique values**
Diagnosis	58929	2789
Procedures	52243	1285

Table 2. Statistics for diagnosis and procedures tables with 1 sequence number

From the literature, predicting top 10 and top 50 codes for both diagnosis and procedures tables are discussed (Huang et al., 2018). Hence, top 10 and top 50 more prevalent codes for both diagnosis and procedures were identified from the datasets containing codes only with sequence 1. Basing on these codes, filtering was performed on the whole datasets of 1.7M and 1.8M which were generated earlier. Table 3 shows the size of datasets after filtering with the overall percentage of the data that has been chosen for final analysis from the note-events dataset considering top 10 and top 50 prevalent codes from diagnosis and procedures table. Therefore, we have got 4 datasets for the final analysis; diagnosis top 10 and diagnosis top 50 prevalent codes; procedures top 10 and procedures top 50 prevalent codes.

Category	No of rows	Unique values	Note-events (%)
Top_10			
Diagnosis	677738	10	32.5
Procedures	632994	10	30.3
Top_50			
Diagnosis	1058988	50	50.8
Procedures	1215197	50	58.3

Table 3. Statistics for diagnosis and procedures tables with top 10 and top 50 prevalent codes

The filtered datasets will be split into 80:20 for training and testing.

4.2 **Feature extraction**

All the 4 datasets were processed to extract features in the following way: As each dataset is having different number of labels (10 for top 10 datasets and 50 for top 50 datasets), all the labels were converted to numeric ranging from 1 to 50 using a function called 'labelencoder' in sklearn. These datasets were divided in 80:20 train and test which were written to a classification model folder. As the language model needs only the text without the labels, all the labels were made to zero which were written to a language model folder. Therefore, classification model folder contains information that we use to create a classifier model. Likewise, the language model folder contains information needed to create a language model.

Once after writing each dataset to the respective folders, 'fixup' functions were defined using python inbuilt syntaxes to clean the text, remove stop words, alpha-numeric characters and punctuations. After cleaning the text, the remaining text was tokenized using a tokenizer available in fastai which can support multi-processing that was built on top of spaCy. Vocabulary is the list of all the token created. All the tokens were then saved to the language model path.

We used term frequency – inverse document frequency (TF-IDF) for feature extraction. TF-IDF evaluates the importance of a word in a document or a collection of document. It is the result of the product of TF and IDF. Term frequency is the number of times a word occurs in each text. While inverse document frequency is number of times a word occurs in the whole document. Words which are used more frequently are given less weight when compared to infrequent ones. Once after tokenizing and counting the frequency of each words (TF) and finally multiplying each word with its corresponding IDF gave the TF-IDF. The two TF-IDF configurations used are: one with top 60,000 words and with a minimum frequency of 2. All these tokens were converted to integers using a torch-text function called 'itos'. Along with these 60,000 tokens two more tokens were inserted: one for unknown (_unk_) and one for padding (_pad_). With these two tokens, total tokens accounted for 60,002. A dictionary was created which can map the integer back to the string. This dictionary doesn't cover everything as this was limited to 60,000 words. If there is some other word that is not present in the dictionary, the dictionary will return zero. This dictionary is called as 'stoi' which can call for each word in the sentence. Using the 'stoi' dictionary all the tokens in the training and the test datasets were converted to numeric's. These training and validation tokens of language model datasets were saved to language model folder using 'np.save' function. Likewise, all the text in classification dataset were processed, converted to tokens, created dictionary mapping string to integer and saved to the classification model folder.

4.2.1 Language model

We use a technique called transfer learning to create our language model. We are training the language model which starts with the weights from the wikitext103 language model trained by Jeremy. In order to load the wikitext103 weights using torch.load, we made sure that we have got the same number of hidden and normal layers and embedding sizes as

that of wikitext103 model. To map the vocabulary from wikitext103 to that of our datasets vocabulary we use the simple 'itos' function that is available for wikitext103 model.

Now we have got new set of weights which are zeros with vocabulary size by embedding size. We then equated the words in our dataset with that of in the wikitext103 model and given labels. In the language model, we predict the next word based on the set of given words. For this, we setup a bunch of dropout values. We created a model data object and the model is transferred to learner using 'learner.fit'. we first trained for single epoch on the last layer to get the new embedding weights. Later we started few epochs of the full model. We then saved the encoder and plotted the loss. All the language models were trained for 3 epochs each.

4.2.2 Classifier

Tokens for the classification data was generated like that of the language model. Hyper parameters were constructed similarly but the dropouts were changed. For the classifier, we classified each text to one of the 10 or 50 classes based on the dataset. We now pass the dataset to the data loader constructor, which gives the batch of that. We then call get_rnn_classifier which creates an RNN encoder and a pooling linear classifier. We then added the number of hidden layers, dropouts and discriminated learning rates for each layer to get more accuracy. We started training with the last layer and found accuracy. We then unfree-zed one more layer and found accuracy. Then we fine-tuned the whole model and trained top 10 diagnosis and procedures datasets for 10 epochs and top 50 diagnosis for 9 epochs and top 50 procedures dataset for 6 epochs to get the accuracy.

4.3 Deep Neural Network Model

ASG Weight-Dropped Long Short-Term Memory (AWD-LSTM)

"Recurrent neural networks (RNNs), such as long short-term memory networks (LSTMs), serve as a fundamental building block for many sequence learning tasks, including machine

translation, language modeling, and question answering. In this paper, we consider the specific problem of word-level language modeling and investigate strategies for regularizing and optimizing LSTMbased models. We propose the weight-dropped LSTM which uses DropConnect on hidden-to-hidden weights as a form of recurrent regularization. Further, we introduce NT-ASGD, a variant of the averaged stochastic gradient method, wherein the averaging trigger is determined using a non-monotonic condition as opposed to being tuned by the user. Using these and other regularization strategies, we achieve state-of-the-art word level perplexities on two data sets: 57.3 on Penn Treebank and 65.8 on WikiText-2. In exploring the effectiveness of a neural cache in conjunction with our proposed model, we achieve an even lower state-of-the-art perplexity of 52.8 on Penn Treebank and 52.0 on WikiText-2" (Stephen et al., 2017, pg. 01).

4.4 Metrics

Different performance metrics like precision, recall, accuracy, f-1 score, ROC curve and AUC value were used to evaluate different datasets, feature extraction methods and the language and classifier models.

Metric	**Formula**
Accuracy	TP+FN/TP+FP+FN+TN
Precision	TP/TP+FP
Recall	TP/TP+FN
F-1	2*(Recall*Precision)/(Recall+Precision)

Table 4. Calculation of different metrics

Where TP – true positives, TN – true negatives, FP – false positives, FN – false negatives.

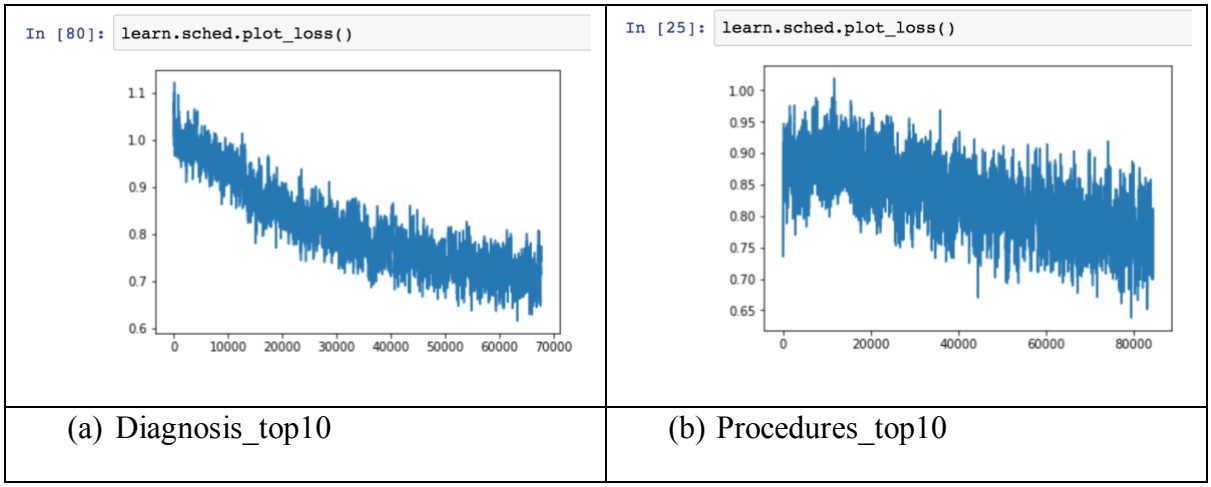
5. Results

This section illustrates the performance of the model:

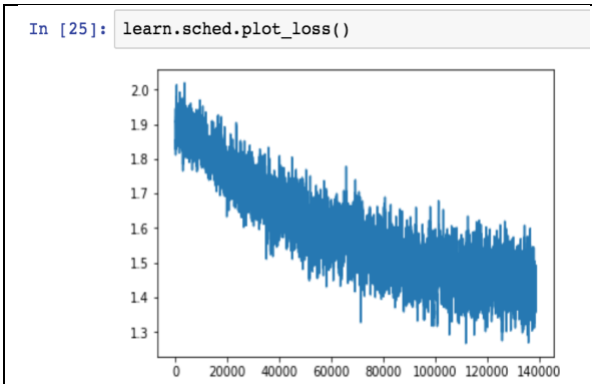
Table 5 demonstrates the performance of each model with respect to their datasets like top 10 and top 50 for diagnosis and procedures respectively. The performance of top 10 models for diagnosis and procedures is relatively better with accuracy above 80% than performance of top 50 models with diagnosis and procedures accuracy of 70.7% and 63.0% accuracy. The precision and recall scores for top 10 datasets are also relatively better than their counter part i.e., top 50 datasets.

S.No	Category	No of rows	Accuracy (%)	Precision	Recall	F-1 Score
1.	Diagnosis_top10	677738	80.3	0.67	0.67	0.66
2.	Procedures_top10	632994	80.5	0.69	0.69	0.69
3.	Diagnosis_top50	1058988	70.7	0.58	0.56	0.55
4.	Procedures_top50	1215197	63.9	0.50	0.50	0.48

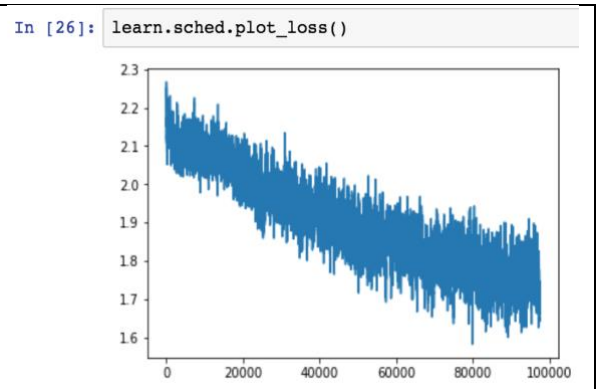
Table 5. Performance of different models



(a) Diagnosis_top10 (b) Procedures_top10



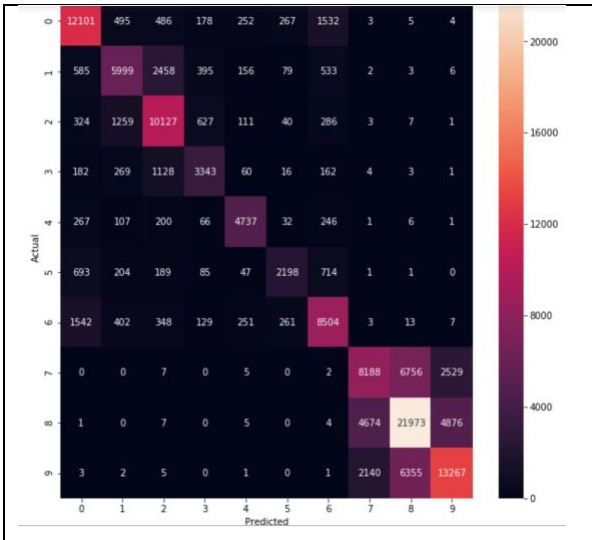
(c) Diagnosis_top50



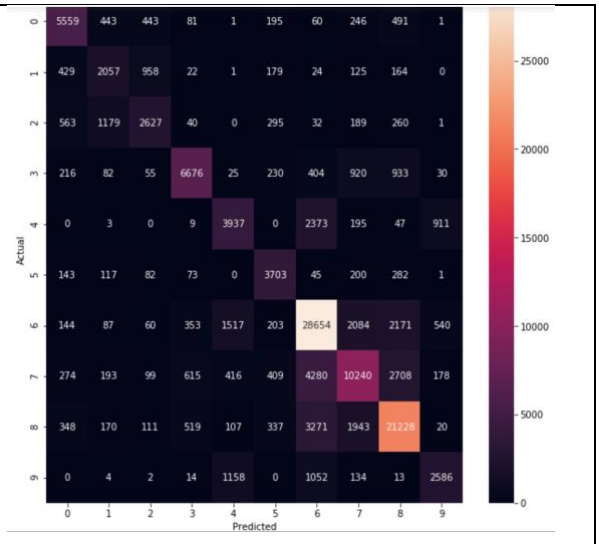
(d) Procedures_top50

Figure 2. Plotting loss after each epoch

From figure 2, we can infer that the loss is reducing for each epoch. Although, magnitude of loss is high in all the datasets, loss is less in top 10 diagnosis and procedures datasets compared to its top 50 counterparts. Loss is top 50 datasets is almost double when compared to their top 10 counterparts.



(a) Diagnosis_top10



(b) Procedures_top10

Figure 3. Confusion matrix

Figure 3 shows the values that are predicted with respect to their actual label. From diagnosis top 10 we can see that the last 3 diagnosis were deviated more towards other labels than other

diagnosis. However, in procedures top 10 classification, all the diagnosis were accurately predicted with few deviations.

```
In [51]: from sklearn.metrics import classification_report
         print(classification_report(actual_values, predicted_values))
```

	precision	recall	f1-score	support
0	0.77	0.79	0.78	15323
1	0.69	0.59	0.63	10216
2	0.68	0.79	0.73	12785
3	0.69	0.65	0.67	5168
4	0.84	0.84	0.84	5663
5	0.76	0.53	0.63	4132
6	0.71	0.74	0.73	11460
7	0.55	0.47	0.50	17487
8	0.63	0.70	0.66	31540
9	0.64	0.61	0.62	21774
micro avg	0.67	0.67	0.67	135548
macro avg	0.70	0.67	0.68	135548
weighted avg	0.67	0.67	0.66	135548

(a) Diagnosis_top10

```
In [49]: from sklearn.metrics import classification_report
         print(classification_report(actual_values, predicted_values))
```

	precision	recall	f1-score	support
0	0.72	0.74	0.73	7520
1	0.47	0.52	0.50	3959
2	0.59	0.51	0.55	5186
3	0.79	0.70	0.74	9571
4	0.55	0.53	0.54	7475
5	0.67	0.80	0.73	4646
6	0.71	0.80	0.75	35813
7	0.63	0.53	0.57	19412
8	0.75	0.76	0.75	28054
9	0.61	0.52	0.56	4963
micro avg	0.69	0.69	0.69	126599
macro avg	0.65	0.64	0.64	126599
weighted avg	0.69	0.69	0.69	126599

(b) Procedures_top10

micro avg	0.56	0.56	0.56	211798
macro avg	0.65	0.52	0.57	211798
weighted avg	0.58	0.56	0.55	211798

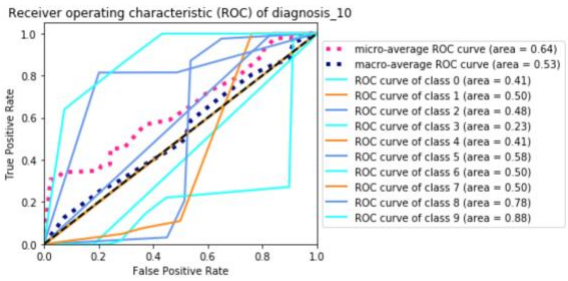
(c) Diagnosis_top50

micro avg	0.50	0.50	0.50	243040
macro avg	0.53	0.39	0.42	243040
weighted avg	0.50	0.50	0.48	243040

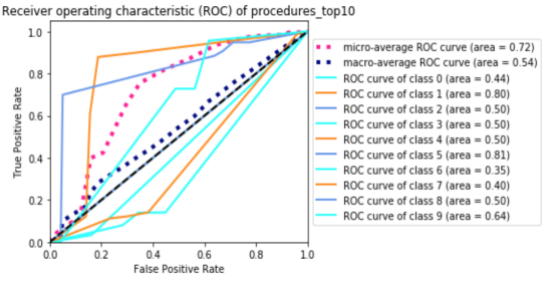
(d) Procedures_top50

Figure 4. Precision, recall and f-1 scores

Figure 4 illustrated the precision, recall and f-1 score for each classifier built. Classifier with top 10 diagnosis and procedures are having higher scores when compared to classifier with top 50 diagnosis and procedures.



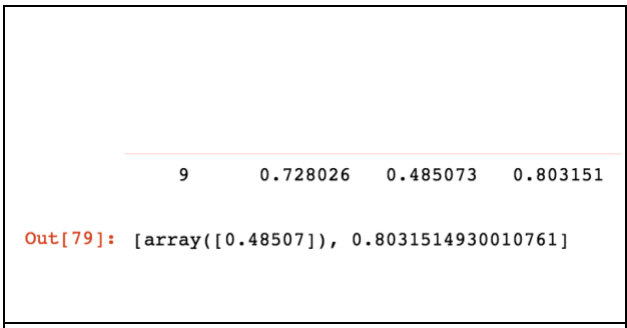
(a) Diagnosis_top10



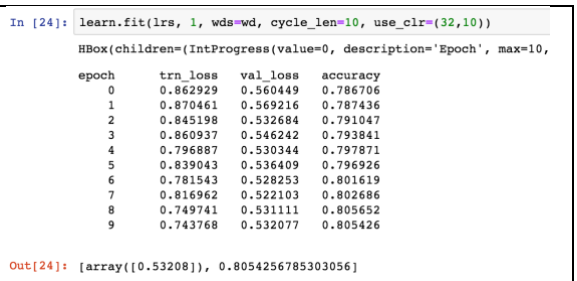
(b) Procedures_top10

Figure 5. ROC curve

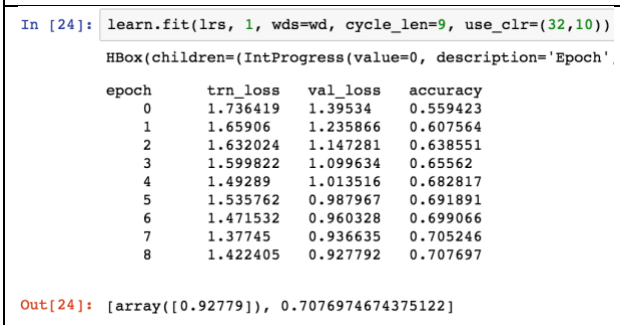
From figure 5 we can see the ROC curve and its area. The overall area for diagnosis top 10 and procedures top 10 classifiers are 64% and 72% respectively. This is because few classes were performing poor, well under their baseline of 50%.



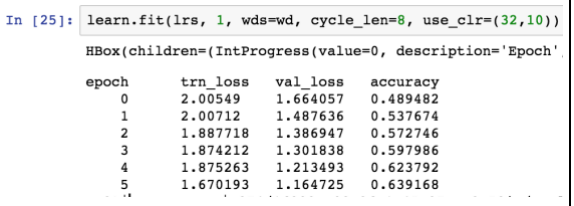
(a) Diagnosis_top10



(b) Procedures_top10



(c) Diagnosis_top10



(d) Procedures_top10

Figure 6. Accuracy of the classifier models

Figure 6 depicts the accuracy of various classifiers. Training and validation loss for diagnosis and procedures top 10 dataset is relatively less i.e., 0.3 and 0.2 respectively compared to their counterparts which are 0.5 respectively. The closer the loss between training and validation datasets, the more accurate the model would be.

Category	**Language model**		**Classifier model**	
	Number of epochs	Time taken (hours)	Number of epochs	Time taken (hours)

Diagnosis_top10	3	11.5	10	31
Procedures_top10	3	10	10	28
Diagnosis_top50	3	14	9	40
Procedures_top50	3	17.5	6	33

Table 6. Summary of language and classifier models with the time taken

Table 6 demonstrates the number of epochs for each model and the time taken for each model.

a) Diagnosis_top10 training 3 epochs on language model took 11.5 hours with accuracy of 68.1%. Training classifier after unfreezing took 31 hours for 10 epochs with the final accuracy of 80.3%. Procedures_top10 training 3 epochs on language model took 10 hours with accuracy of 65.7%. Training classifier for 10 epochs took 28 hours with accuracy of 80.5%.

b) Diagnosis_top50 training 3 epochs on language model took 14 hours with accuracy of 65.5%. Training classifier after unfreezing took 40 hours for 9 epochs with the final accuracy of 70.7%. Procedures_top50 training 3 epochs on language model took 17.5 hours with accuracy of 51.0%. Training classifier for 6 epochs took 33 hours with accuracy of 63.9%.

As these were performed on different GPU's with varied performance, the time taken is not directly related to the size of the dataset.

6. Conclusion

In this study, we observed the performance of novel variant of LSTM i.e., AWD-LSTM on MIMIC-III discharge summaries. The models use the deep learning NLP techniques which assign ICD-9 code automatically to the clinical text. The AWD-LSTM model for predicting top 10 diagnosis and procedures are better than top 50 codes of diagnosis and procedures. The accuracy of diagnosis and procedures top 10 models are 80.3% and 80.5% respectively. The

accuracies of top 50 diagnosis and procedures are 70.7% and 63.9% respectively. we hope our implementation of AWD-LSTM will serve as a baseline for further research in identifying diagnosis, procedures and treatments using a single model at once.

7. Future work

The uniqueness of this study lies in prediction of a diagnosis and procedures from the clinical text. Although many studies reported the coding of diagnosis from the discharge notes, taking entire note for each admission and predicting diagnosis and procedures is the innovation of the study. Future work relies on the prediction of diagnosis, procedures and treatment using a single model unlike that of the multiple models predicting diagnosis and procedures distinctly. Also, it needs to be refined to enhance the accuracy and performance of the model when more than 10 diagnosis would be employed.

8. Challenges

The primary challenge is the availability of the resources. Although the resources are available for small scale operations handling 800GB data require huge amount of resources and time. However, the school GPU is available from the beginning of the project, as it is a shared resource, all the GPU's not available half the time to train our model. Moreover, the Google resources are associated with high costs. Even though $600 of free credit has been from Google, running on a large scale would have required lot more than the available free credits. The Quadro P6000 is a new GPU that is made available by my mentor for this project to move forward in this limited resources environment. It is a grant from my mentor as part of another project.

S.No.	GPU	Location	Hours of operation
1.	GetForce GTX 1080	School	430

2.	Quadro P6000	School	140
3.	NVIDIA Tesla V 100	Google	260

Table 7. Hours of operation of various GPU's

9. References

Adkins D. E. (2017). Machine Learning and Electronic Health Records: A Paradigm Shift. *The American journal of psychiatry*, *174*(2), 93-94.

Artetxe, A., Beristain, A., & Grana, M. (2018). Predictive models for hospital readmission risk: A systematic review of methods. *Comput Methods Programs Biomed, 164*, 49-64. doi:10.1016/j.cmpb.2018.06.006

Beaulieu-Jones, B. K., Orzechowski, P., & Moore, J. H. (2018). Mapping Patient Trajectories using Longitudinal Extraction and Deep Learning in the MIMIC-III Critical Care Database. *Pac Symp Biocomput, 23*, 123-132.

Bibault, J. E., Giraud, P., & Burgun, A. (2016). Big Data and machine learning in radiation oncology: State of the art and future prospects. *Cancer Lett, 382*(1), 110-117. doi:10.1016/j.canlet.2016.05.033

Bibault, J. E., Giraud, P., Housset, M., Durdux, C., Taieb, J., Berger, A., . . . Burgun, A. (2018). Deep Learning and Radiomics predict complete response after neo-adjuvant chemoradiation for locally advanced rectal cancer. *Sci Rep, 8*(1), 12611. doi:10.1038/s41598-018-30657-6

Choi, E., Schuetz, A., Stewart, W. F., & Sun, J. (2017). Using recurrent neural network models for early detection of heart failure onset. *J Am Med Inform Assoc, 24*(2), 361-370. doi:10.1093/jamia/ocw112

Dubois, S., & Romano, N. (n.d.). Learning Effective Embeddings from Medical Notes, 10.

Gehrmann, S., Dernoncourt, F., Li, Y., Carlson, E. T., Wu, J. T., Welt, J., . . . Celi, L. A. (2018). Comparing deep learning and concept extraction based methods for patient phenotyping from clinical narratives. *PLOS ONE, 13*(2), e0192360. doi:10.1371/journal.pone.0192360

Geraci, J., Wilansky, P., de Luca, V., Roy, A., Kennedy, J. L., & Strauss, J. (2017). Applying deep neural networks to unstructured text notes in electronic medical records for phenotyping youth depression. *Evidence Based Mental Health, 20*(3), 83-87. doi:10.1136/eb-2017-102688

Huang, Z., Dong, W., Duan, H., & Liu, J. (2018). A Regularized Deep Learning Approach for Clinical Risk Prediction of Acute Coronary Syndrome Using Electronic Health Records. *IEEE Trans Biomed Eng, 65*(5), 956-968. doi:10.1109/tbme.2017.2731158

Hung, C. Y., Lin, C. H., & Lee, C. C. (2018). Improving Young Stroke Prediction by Learning with Active Data Augmenter in a Large-Scale Electronic Medical Claims Database. *Conf Proc IEEE Eng Med Biol Soc, 2018*, 5362-5365. doi:10.1109/embc.2018.8513479

Jiwani, A., Himmelstein, D., Woolhandler, S., & Kahn, J. G. (2014). Billing and insurance-related administrative costs in United States' health care: synthesis of micro-costing evidence. *BMC health services research, 14*, 556-556. doi:10.1186/s12913-014-0556-7

Korvigo, I., Holmatov, M., Zaikovskii, A., & Skoblov, M. (2018). Putting hands to rest: efficient deep CNN-RNN architecture for chemical named entity recognition with no hand-crafted rules. *J Cheminform, 10*(1), 28. doi:10.1186/s13321-018-0280-0

Li, F., Liu, W., & Yu, H. (2018). Extraction of Information Related to Adverse Drug Events from Electronic Health Record Notes: Design of an End-to-End Model Based on Deep Learning. *JMIR Med Inform, 6*(4), e12159. doi:10.2196/12159

Li, F., Zhao, C., Xia, Z., Wang, Y., Zhou, X., & Li, G. Z. (2012). Computer-assisted lip diagnosis on Traditional Chinese Medicine using multi-class support vector machines. *BMC Complement Altern Med, 12*, 127. doi:10.1186/1472-6882-12-127

Liang, H., Sun, X., Sun, Y., & Gao, Y. (2017). Text feature extraction based on deep learning: a review. *EURASIP journal on wireless communications and networking, 2017*(1), 211-211. doi:10.1186/s13638-017-0993-1

Liang, Z., Liu, J., Ou, A., Zhang, H., Li, Z., & Huang, J. X. (2018). Deep generative learning for automated EHR diagnosis of traditional Chinese medicine. *Comput Methods Programs Biomed*. doi:10.1016/j.cmpb.2018.05.008

Liu, Z., Yang, M., Wang, X., Chen, Q., Tang, B., Wang, Z., & Xu, H. (2017). Entity recognition from clinical texts via recurrent neural network. *BMC Med Inform Decis Mak, 17*(Suppl 2), 67-67. doi:10.1186/s12911-017-0468-7

Lyu, C., Chen, B., Ren, Y., & Ji, D. (2017). Long short-term memory RNN for biomedical named entity recognition. *BMC Bioinformatics, 18*(1), 462. doi:10.1186/s12859-017-1868-5

Merity, S., Keskar, N. S., & Socher, R. (2017). Regularizing and optimizing LSTM language models. *arXiv preprint arXiv:1708.02182*.

Miotto, R., Li, L., Kidd, B. A., & Dudley, J. T. (2016). Deep Patient: An Unsupervised Representation to Predict the Future of Patients from the Electronic Health Records. *Sci Rep, 6*, 26094. doi:10.1038/srep26094

Nguyen, P., Tran, T., Wickramasinghe, N., & Venkatesh, S. (2016). Deepr: A Convolutional Net for Medical Records. *CoRR, abs/1607.07519*.

Reddy, B. K., & Delen, D. (2018). Predicting hospital readmission for lupus patients: An RNN-LSTM-based deep-learning methodology. *Comput Biol Med, 101*, 199-209. doi:10.1016/j.compbiomed.2018.08.029

Shickel, B., Tighe, P. J., Bihorac, A., & Rashidi, P. (2018). Deep EHR: A Survey of Recent Advances in Deep Learning Techniques for Electronic Health Record (EHR) Analysis. *IEEE Journal of Biomedical and Health Informatics, 22*(5), 1589-1604. doi:10.1109/JBHI.2017.2767063

Wu, Y., Jiang, M., Xu, J., Zhi, D., & Xu, H. (2017). Clinical Named Entity Recognition Using Deep Learning Models. *AMIA Annu Symp Proc, 2017*, 1812-1819.

Xiao, C., Choi, E., & Sun, J. (2018). Opportunities and challenges in developing deep learning models using electronic health records data: a systematic review. *J Am Med Inform Assoc, 25*(10), 1419-1428. doi:10.1093/jamia/ocy068

Yang, Z., Huang, Y., Jiang, Y., Sun, Y., Zhang, Y.-J., & Luo, P. (2018). Clinical Assistant Diagnosis for Electronic Medical Record Based on Convolutional Neural Network. *Scientific Reports, 8*(1), 6329. doi:10.1038/s41598-018-24389-w

Zhu, Q., Li, X., Conesa, A., & Pereira, C. (2018). GRAM-CNN: a deep learning approach with local context for named entity recognition in biomedical text. *Bioinformatics, 34*(9), 1547-1554. doi:10.1093/bioinformatics/btx815

Poster

SCHOOL OF INFORMATICS AND COMPUTING
INDIANA UNIVERSITY
IUPUI

Natural Language Processing of Clinical Notes for Identifying Diagnosis and Procedures using Neural Networks

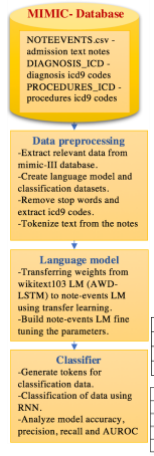
Siddhartha Nuthakki, Pharm D, Judy Gichoya, MD MS, Saptarshi Purkayastha, PhD

Introduction

- Coding diagnosis and procedures in medical records is a critical process in the healthcare industry.
- In the US, Billing and Insurance-related (BIR) activities costed around $471 billion in 2012 (Jiwani et al., 2014) which accounts for 25% of all the U.S. hospitals spending .
- Coding is important at various levels, from creating accurate billing, getting reimbursement from payers and generating a standardized patient care record.
- The objective of the study is to build a natural language processing model which can map clinical notes to medical codes and predict the final diagnosis from unstructured entries like history of present illness, symptoms at the time of admission.
- We employed the state-of-the-art deep learning method ASG Weight-Dropped Long Short-Term Memory (AWD-LSTM) on the largest publicly shared emergency department clinical notes dataset MIMIC III.

Previous work

- MIMIC-III contains de-identified clinical data of over 53,000 hospital admissions for adult patients to the intensive care units (ICU) at the Beth Israel Deaconess Medical Center from 2001 to 2012.
- The dataset comprises several types of clinical notes: discharge summaries (n = 52,746) and nursing notes (n = 812,128).
- Gehrmann et al., 2018 showed that state-of-the-art deep learning methods outperformed traditional concept extraction based methods.
- Huang et al., 2018 focused on detection of diagnosis codes from discharge summaries and were able to achieve 89.67% accuracy using RNN and CNN.



Methodology

- Data from multiple tables were integrated using the Python Dask parallel processing library.
- After merging all events, procedures, diagnosis and clinical notes, the data became unmanageable to 800 GB, and thus the project had to be rescoped.
- The MIMIC III notes are long with multiple structured subsections, and we used the full-texts, but only used the primary diagnosis and main procedure for the admission as labels.

Category	No of rows	Unique values
Note-events	2083180	2023185
Diagnosis	651047	6984
Procedures	240095	2009

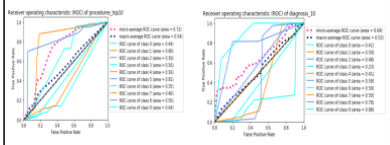
Category	Population	Note-events (%)
Diagnosis_top10	677738	32.5
Procedures_top10	632994	30.3
Diagnosis_top50	1058988	50.8
Procedures_top50	1215197	58.3

Results

- Our models predicted the top-10 and top-50 ICD-9 codes of diagnosis with 80.3% and 70.7% accuracy respectively.
- The second predicted the top-10 and top-50 ICD-9 codes of procedures with 80.5% and 63.9% accuracy respectively.

S.No	Category	Accuracy (%)	Precision	Recall	F-1 Score
1.	Diagnosis_top10	80.3	0.67	0.67	0.66
2.	Procedures_top10	80.5	0.69	0.69	0.69
3.	Diagnosis_top50	70.7	0.58	0.56	0.55
4.	Procedures_top50	63.9	0.50	0.50	0.48

Results continued



Conclusion

- With the scores from the present models, the next step is to deploy on a small-scale real-world scenario and compare with human coders as gold standard.
- Prediction of final diagnosis from either history of present illness or symptoms can let physicians identify promising treatment, which can potentially transform the conventional care of diagnosis followed by treatment.

Challenges

- Labeled gold standard
- Hardware resources

S.No	GPU	Location	Hours of operation
1.	GetForce GTX 1080	School	430
2.	Quadro P6000	School	140
3.	NVIDIA Tesla V 100	Google	260

References and Acknowledgements

Work on the project was partially supported through the Nvidia GPU grant program to Purkayastha (2018)

[1] Jiwani, A., Himmelstein, D., Woolhandler, S., & Kahn, J. G. (2014). Billing and insurance-related administrative costs in United States' health care: synthesis of micro-costing evidence. *BMC health services research, 14*, 556-556. doi:10.1186/s12913-014-0556-7

[2] Huang, J., Osorio, C., & Sy, L.W. (2018). An Empirical Evaluation of Deep Learning for ICD-9 Code Assignment using MIMIC-III Clinical Notes. *CoRR, abs/1802.02311*.

[3] Pollard, T. J. & Johnson, A. E. W. The MIMIC-III Clinical Database http://dx.doi.org/10.13026/C2XW26 (2016).