DECEMBER 12, 2018

ILLINOIS INSTITUTE OF TECHNOLOGY,
CHICAGO

AUTOMATIC TAGGING OF CLINICAL BASED ON ICD – 10 CM

CS – 594 RESEARCH PROBLEM

Submitted by: Abhishek Bhardwaj
Advised by: Prof. Sanjiv Kapoor

1. INTRODUCTION

1.1. Problem Definition

The World Health Organization maintains a healthcare classification system known as the International Classification of Disease[i]. This classification provides a map to various health conditions. It is the most widely used healthcare classification system, providing a system of diagnostic codes and their description, used for reporting diseases and health conditions, assisting in medical billing, collecting morbidity and mortality data etc.[ii] These codes are often used in not just billing purposes but also for research. In Public Health industry, the use of these standardized codes helps to assimilate healthcare crisis and emergency. The use of such standardized data allows proper and effective sharing of data. While ICD is so important in recording clinical data, and making financial decisions, assigning these codes to clinical documentation is not easy and often requires extensive technical writing and hence involves higher labor costs. The process of manual entry of these codes are time-consuming and subjective to errors. This process becomes even more difficult when, for writing diagnostic descriptions, physicians use abbreviations and synonyms which can cause ambiguity and errors in matching ICD codes to their respective labels. Furthermore, several diagnosis descriptions are so closely related and are required to be classified as a combination of the codes, however, the unexperienced coders often fail to do so and hence result in unbundling. Lastly, the ICD codes are arranged in a hierarchical structure with the more generalized codes at the top and the more specific codes at the bottom. Most of the time a miscoding can occur when a coder does not go through the entire list and assigns a generic code instead of a specific one[iii]. These coding errors not only result in clinical mistakes but puts a hole through financial stability in the USA[iv].

To reduce coding errors and cost, this paper aims at building an ICD coding mechanism which automatically and accurately translates the diagnosis descriptions into ICD codes. Medical coding is a major facet in maintaining patient records and in obtaining healthcare reimbursements. It takes the descriptions of diseases, injuries, and healthcare procedures from the health care provider and transforms it into a numeric or alphanumeric code to accurately describe the diagnosis of the procedures performed. This paper leverages the increasing prominence of electronic health records (EHR), and utilizes natural language processing (NLP) algorithms to support the process of medical coding. This paper aims to provide a solution to the issue, by automatically inferring the respective codes from the data in the EHR.

1.2. ICD – 10 Procedure Coding System

The 2016 edition of the ICD -10 codes is divided into 21 chapters, based about the codes each chapter contains. With the advancement of ICD – 9 to the more complex ICD – 10 codes; a need for effective coding methods becomes especially acute. While, the earlier version of ICD – 9 codes contained only 14,440 codes, the ICD – 10 consists of 68,000 codes with addition of another 368 codes effective 1st October 2011.

The ICD is a system used by healthcare providers to classify and code all diagnosis, symptoms and procedures recorded in conjugation with hospital care in the United States. It is based on the international classification of diseases by World Health Organization (WHO). The first character of the code is an alpha character excluding “u”, the second to 7th character is alpha numeric. The first three characters categorize the injury and the fourth through sixth characters describe in greater detail the cause, anatomical location and severity of an injury or illness. The seventh character is an extension digit and used to classify an initial, subsequent or sequela (late effect) treatment encounter.

Such complex structure of ICD codes over emphasizes the need for automatic coding to reduce human error. Nevertheless, the attempts to predict the codes as unitary entities are bound to suffer data sparsity problems even with a large training corpus. Due to the complex nature of the code structure, defining approximately 70,000 codes can limit the scope; as to how much can be established by matching the code descriptions and data in the EHR.

1.3. Related Work

The possibilities of automating the ICD coding task have been studied extensively since a very long time. A popular approach in the literature over the last several years has been to use NLP to extract codes mapped to controlled sources from text. Goldstein et al.[v] evaluates three systems for automatically predicting ICD – 9 Codes from short excerpts of text, resulting in, semantic information significantly contributing to ICD-9-CM coding with lexical elements. According to them, hand – crafted semantic information system in conjugation with lexical elements results in algorithmically outperforming more complex systems. Rios and Kavuluru[vi] conduct experiment on EMR dataset curated from the University of Kentucky Medical Center to show that feature selection, training data selection, and probabilistic thresholding provide significant gains in performance. Farkas and Szarvas[vii] demonstrate that successful ICD-9-CM coding can be reproduced by replacing several laborious steps with machine learning models. These models preserve the favorable aspects of rule-based classifiers, like good performance, and their development can be achieved rapidly and requires less human effort and hence lesser chances of errors. Thus, development of such systems can be feasible for a more labeled data. Lima et al[viii]. utilized hierarchical structure of the ICD-9 code set, a property that is less useful when only a limited number of codes is used, as in our study. Crammer et al[ix]. also, described a multi-component coding system of radiology reports using machine learning, a rule-based system, and

an automatic coding system based on human coding policies. Baud et al[x]. detail an overview of the problems in the task of ICD-10 encoding using the ICD – 9 coding approaches. Over the past few decades, information extraction from free text EHR documents has advanced drastically. Improvements in system performance will subsequently enhance the acceptance and usage of automatic ICD coding in clinical contexts.

Latent Dirichlet Allocation(LDA) and Latent Semantic Analysis(LSA) are two widely used algorithms in the arena of automatic ICD coding and topic modelling. Researchers and academicians often modify these algorithms based on their need. One similar approach is being utilized by Thomas Hofmann[xi] in his work where he describes a Probabilistic LSA, with real world implications in information retrieval and filtering, natural language processing, and machine learning from text. PLSA is a latent variable model for co-occurrence data which associates unobserved class variable with each observation. The experimental analysis of this approach, performed by the author, focuses on two tasks perplexity minimization and automated indexing of documentation. The experiments consistently validate the advantages of the PLSA, resulting in performance gains for all the data sets used in the experiment. Not only LDA and LSA but many other methods are also being used in the industries and by researchers to classify and tag similar documents. Researchers after performing basic natural language procedures (NLP) like tokenization, stop word removal, stemming construct document term matrix to have a vector representation of documents. Another team of researchers lead by David M. Blei et.al[xii] in their work explains the major drawback of the topic modeling, i.e. scalability, and aims to provide a solution for large scale hierarchical topic modeling(HTC). Their research uses an existing parallel implementation of Latent Dirichlet Allocation (LDA) to provide a scalable mechanism for learning hierarchies from large and complex data sets. The results of this approach assess the run time and

human evaluations of quality on large data sets. The authors' HTC modeling approach achieves its aim of scalability which is shown by the results of the two large scale document repositories.

2. METHODOLOGY

Figure 1 shows the overview of the approach taken in this paper. The task of automating ICD-10 coding involves data processing, feature extraction, and model training and testing. This paper focuses on establishing a system that can tag the clinical text with their respective ICD -10-CM codes.

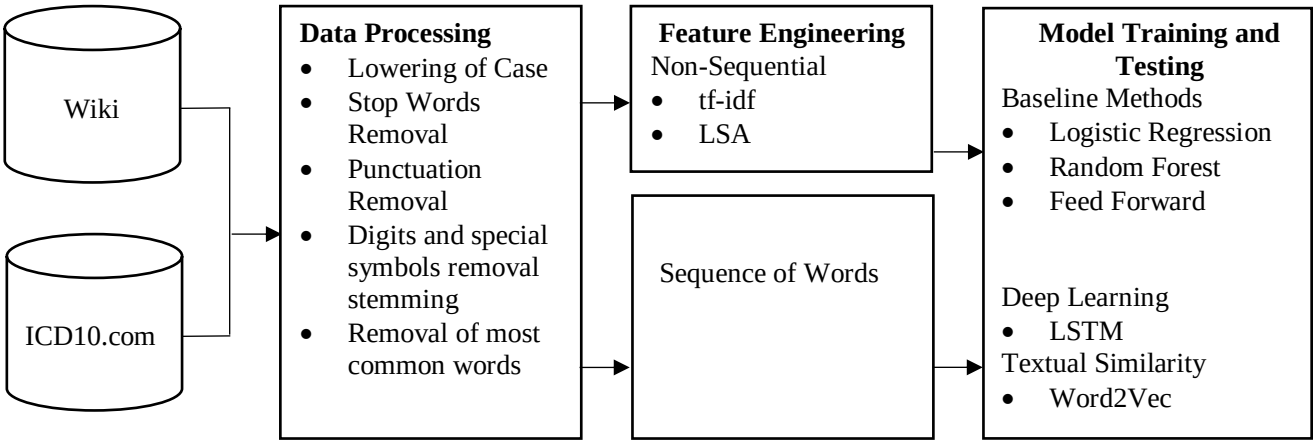

Figure 1. Methodology pipeline flow

2.1. Data

To get a good training data for ICD-10 we crawled the data from two most reliable sources on is the Wikipedia directory and second is https://icd10.com which is the place where the ICD-10 codes are described. ICD codes are divided into 22 chapter with each chapter have blocks with children codes. For example, Chapter-1 named contained 200 codes marked from A00-B-99. Table 2.1 shows the complete list of chapters and the subsequent code ranges along with their titles.

Table 1. ICD-10 Chapters List

Chapter	Blocks	Title
I	A00–B99	Certain infectious and parasitic diseases
II	C00–D48	Neoplasms
III	D50–D89	Diseases of the blood and blood-forming organs and certain disorders involving the immune mechanism
IV	E00–E90	Endocrine, nutritional and metabolic diseases
V	F00–F99	Mental and behavioural disorders
VI	G00–G99	Diseases of the nervous system
VII	H00–H59	Diseases of the eye and adnexa
VIII	H60–H95	Diseases of the ear and mastoid process
IX	I00–I99	Diseases of the circulatory system
X	J00–J99	Diseases of the respiratory system
XI	K00–K93	Diseases of the digestive system
XII	L00–L99	Diseases of the skin and subcutaneous tissue
XIII	M00–M99	Diseases of the musculoskeletal system and connective tissue
XIV	N00–N99	Diseases of the genitourinary system
XV	O00–O99	Pregnancy, childbirth and the puerperium
XVI	P00–P96	Certain conditions originating in the perinatal period
XVII	Q00–Q99	Congenital malformations, deformations and chromosomal abnormalities
XVIII	R00–R99	Symptoms, signs and abnormal clinical and laboratory findings, not elsewhere classified
XIX	S00–T98	Injury, poisoning and certain other consequences of external causes
XX	V01–Y98	External causes of morbidity and mortality
XXI	Z00–Z99	Factors influencing health status and contact with health services
XXII	U00–U99	Codes for special purposes

Since many of the ICD codes are very closely related their descriptions were identical for example, P03.2 (Fetes and new-born affected by forceps delivery) and P03.3 (Fetus and new-born affected by delivery by vacuum extractor) had identical descriptions.

2.2. Word Cloud

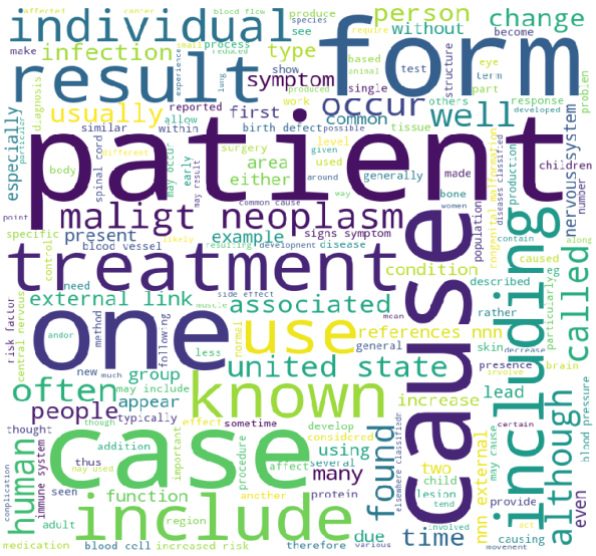

Figure 3. Wikipedia word cloud

Figure 2. ICD10.com word cloud

2.3. Data Pre-processing

Wikipedia and ICD10.com together represent a large dataset relating to classification of diseases. In order to construct the vocabulary and compute the data engineering and topic modelling, firstly, text tokenization is performed and the tokens are filtered in a training dataset. We use the training set to train our models, and to develop a set to routinely validate that we are not overfitting the train set. A document term matrix with the count of each term in each token is created. To tokenize, we split on spaces, disregard certain types of punctuation like periods and semicolons, and remove stop words and numerical tokens. Finally, each term is multiplied with the corresponding IDF. Top 1000 terms with the highest TF –IDF scores are regarded as the final vocabulary for training the model. After constructing the vocabulary, a bag of terms feature vector for each patient token is constructed. We chose a bag of terms representation because the medical tokens do not really have a grammatical structure and while the terms order within the small segments of text matters, the order of each segment does not matter too much.

2.4. Feature Engineering

For dimension reduction, reducing the number of random variables, feature extraction is utilized as the feature engineering method. This paper focuses on the following methods for feature engineering:

2.4.1. Term frequency – inverse document frequency

Term frequency (TF) computes normalized term frequency that is, it measures how frequently a term occurs in a document, whereas, Inverse Document Frequency (IDF) measures how important a term is. Together, TF – IDF are used to evaluate the importance of a word to a document in a collection of documents. While computing TF, all terms are considered equally important. However, it is known that certain terms such as “is”, “the”, “of”, may appear many times but have little importance. Thus, weighing down the frequent words and scaling up the rare terms is important. The following logarithm of IDF is used for our calculations:

$$IDF(t) = \log N / DF(d, t)$$

where N is the total number of documents and DF (d, t) is the number of documents that contain term t.

2.4.2. Latent semantic Analysis

Latent semantic analysis (LSA) takes tf-idf one step further. LSA method is technique in text classification to represent hidden semantic structure of a term document matrices in which rows are documents and columns are tokens. Generally, LSA analyzes relationships between a term and concepts contained in a collection of text. It correlates the semantically related terms that are hidden in the text. LSA uses Singular Value Decomposing (SVD) to identify a pattern between the terms and concepts contained in the text and to find relation between documents. It has ability to extract the conceptual content of a body of text. It overcomes the issue of multiple words having

similar meanings and words that have multiple meanings[xiii]. We utilize LSA in this paper for creating vector representation of document. This allows us to compare different documents by measuring the vector distances, which in turn helps in classifying clinical documents to determine which ICD-10-CM label they belong to. One limitation of LSA is that each word is represented as a single point with the same meaning; therefore, in this representation, polysemes of words cannot be differentiated. Also, the final output of LSA, which consists of axes in Euclidean space, is not interpretable or descriptive[xiv].

3. MODEL TESTING AND TRAINING

Automatic coding of ICD codes involves a multi-label problem. Most of the time when recording patient history of the disease and diagnosis, we see that one patient suffers from more than just one disease. This leads to assigning multiple ICD label to one particular patient.

In this paper, three baseline approaches are adopted: Logistic Regression, Random Forests and Feed-Forward Neural Network. In the baseline approach, problem transformation methods are used to get the multi-label output for Logistic Regression and Random Forest classifiers. Each set of target labels are simply trained on different models for top 100 ICD-10 codes. We restrict ourselves to 100 codes just for identifying the best algorithm and feature engineering pair. Each model independently predicts a mutual exclusive binary output (0 or 1) for each sample data. For Feed-Forward Neural Network, algorithm adaptation based methods are used, since neural network can be easily adapted to multi-label problem by setting up multiple neurons in the network output layer. Similarly, for Feed-Forward Neural Network, we also used algorithm adaptation-based methods to our deep learning models as well.

3.1. Baseline Model Testing

3.1.1. Logistic Regression

As a baseline, we implement a Binomial Logistic Regression model. In this model, a separate logistic regression model is trained for each label (ICD -10 code or category), and each model independently predicts the value (0 or 1) of that label. As input features to this model, we take the sum of all the bag-of-terms token vectors for each patient and normalize them.

3.1.2. Random Forest

This is the second approach used in this paper and utilizes similar approach and input as the logistic regression, that is, one model for each label.

3.1.3. Feed-Forward Neural Network

We then implement a basic feed-forward neural network, to understand the baseline performance of multi label classifications. The input features, once again, are the unnormalized sum of all the bag-of-terms token vectors for each patient. ReLU activation functions for the hidden units is used. In addition, the sigmoid cross entropy as the loss function to optimize was used and to predict labels, sigmoid function on the output layer is applied.

3.2. Deep Learning Models: Recurrent Neural Network - LSTM

Unlike Feed-forward neural networks in which activation outputs are propagated only in one direction, the activation outputs from neurons in Recurrent Neural Networks (RNN) propagate in both directions (from inputs to outputs and from outputs to inputs). This creates loops in the neural network architecture which acts as a 'memory state' of the neuron, which allows the neurons an ability to remember what have been learned so far. Among the sophisticated recurrent units, we utilize the Long Short Term Memory (LSTM) approach. After establishing baseline models on the top 1000 code dataset, we implement a recurrent neural network (RNN) with two latent layers with

100 iterations. First layer is an LSTM layer with 1000 units and it returns sequences. A dropout layer is applied LSTM layer to avoid overfitting of the model. Finally, we have the second layer as a fully connected layer with a 'softmax' activation and neurons equal to the number of unique characters. The model is fit over 100 epochs, with a batch size of 64. We then fix a random seed (for easy reproducibility) and start generating characters. The prediction from the model gives out the character encoding of the predicted character, it is then decoded back to the character value and appended to the pattern. Eventually, after enough training epochs, it gives a final output.

3.3. Textual Similarity - Word2Vec

The word2vec model is used to compare the similarity between the earlier established train data set and the new data set. Word2vec model takes a corpus as a input and produces a vector space. Each unique word in the corpus is represented as a vector and words that share a common context are placed next to each other. The similarity between the two words is measured using cosine similarity.

3.4. Metrics:

There are several types of metrics used in multi-label classification: label-based, sample-based, and ranking-based metrics. This paper utilizes combination of precision, accuracy, F-score and recall metrics. Specifically, the following metrics are used[xv]:

$$Precision = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i \cap Z_i|}{Z_i} \qquad Recall = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i \cap Z_i|}{Y_i}$$

$$F_1 = \frac{1}{n}\sum_{i=1}^{n}\frac{2\,|Y_i \cap Z_i|}{|Y_i| + |Z_i|} \qquad Accuracy = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i \cap Z_i|}{|Y_i \cup Z_i|}$$

where Y_i is the set of predicted labels, Z_i is the set of ground truth labels, and n is the number of samples. q is the number of total samples.

4. RESULTS

This section illustrates the performance as: (1) baseline results, (2) deep learning analysis and (3) textual similarity analysis. The features extracted include non-sequential - tfidf and LSA – these features were used in Logistic Regression, Random Forest, Feed-Forward Neural Network and LSTM. The pre-processed text was directly used in a matrix based on word2vec.

4.1. Model Performance under Different Configurations

Different model configurations are tried to give us insight into the most appropriate model configuration. Table 2. describes the different methods of feature extraction used and the parameters tweaked. The features extracted are divided into 2 categories: non-sequential and sequential. The non-sequential features include tfidf and LSA, which were used in Logistic Regression, Random Forest, and Feed-Forward Neural Network. The sequential features include wordseq which was used in word2vec to generate vectors and for LSTM wordseq was used in conjunction with an embedding matrix. All the feature engineering methods were applied after pre-processing the data which includes stop words, punctuation and symbol removal. We also preformed porter stemming on the data to clean the tokens.

4.2. Best Model Overview

Figure 15 – AUC and F score and Figure 16 – Precision Recall and Accuracy shows the model performance comparison for all the feature engineering techniques with various classification algorithms. Raw data of the results also shown in table 2. Performance matrix

ACCURACY	ALG_NAME	AUC	F-Score	PRESCISION	RECALL	RUNTIME
0.99	RandomForestClassifier_LSA_100	0.5	0.497487437	0.5	0.5	1.835
0.48	FeedForwardClassifier_LSA_100	0.507676768	0.331282054	0.5	0.51	0.633
0.99	SVM_TFidf_100	0.504848485	0.507052055	0.62	0.5	22.392
0.98	LSTM_100	0.595959596	0.595959596	0.6	0.6	303.234
0.07	FeedForwardClassifier_TFidf_100	0.500808081	0.067918276	0.5	0.5	65.093
0.99	Logistic_Regression_LSA_100	0.5	0.497487437	0.5	0.5	0.21
0.83	Word2Vector_ALL	0.798646334	0.458685104	0.5	0.8	7.641
0.01	Logistic_Regression_TFidf_100	0.5	0.00990099	0	0.5	18.268
0.97	RandomForestClassifier_TFidf_100	0.507676768	0.502843734	0.5	0.51	6.258
0.99	SVM_LSA_100	0.499747475	0.497361146	0.49	0.5	0.286

Table 2. Performance matrix

For top-100-code LSTM generated the best f1 results at 0.5959. This makes sense because in clinical data each label has large descriptions and there is not much difference between each label. The baseline models (Logistic Regression, SVM and Random Forest) show good accuracy on testing data. All the Baseline models have performed better with LSA feature engineering as compared to Tfidf. A possible implication could be that uniqueness of each code can be identified better by LSA than Tfidf, as LSA semantically build a vector for each record. In terms of accuracy all the baseline models have outperformed LSTM and Wor2vec, but when we look at AUC and f-1 of baseline models the results are not as convincing. With word2vec and LSTM showing a significant AUC indicates that they have a better performance than the random guessing. Figures 4 – 9, shows the ROC curves of all the models reflecting all the baseline models are very close to the random guessing even after parameter tuning as well. The ROC curve for LSTM and Word2Vec signifies that these models could extract information from the sequence of words, otherwise lost in non-sequential feature extraction, thereby improving the f1 and the AUC results. To finally conclude, Word2Vec due to its simple computing the cosine similarity with vector index out performs all the other algorithms due to presence exact pathogen names in clinical text reducing the cosine angle between them.

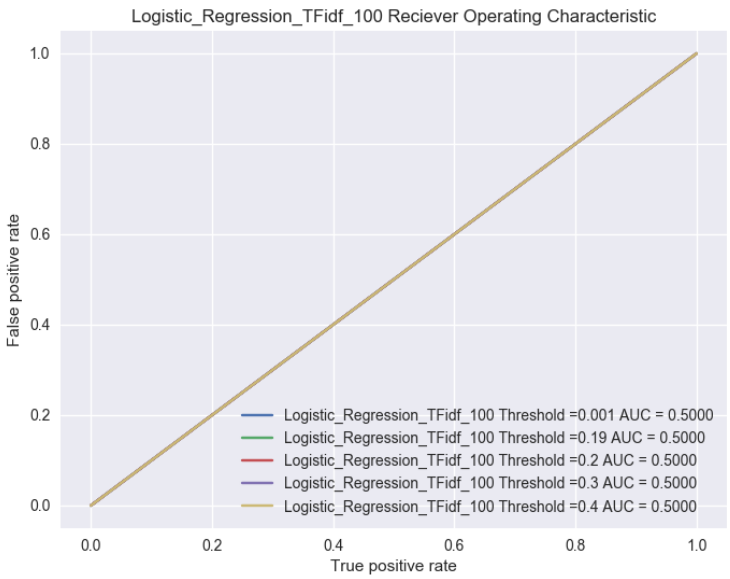

Figure 7 ROC_Logistic Regression on tfidf

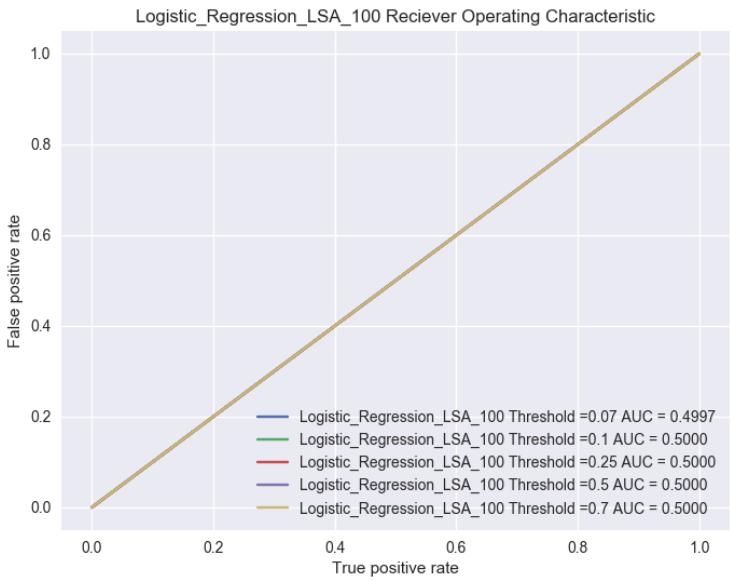

Figure 6 ROC_Logistic Regression on LSA

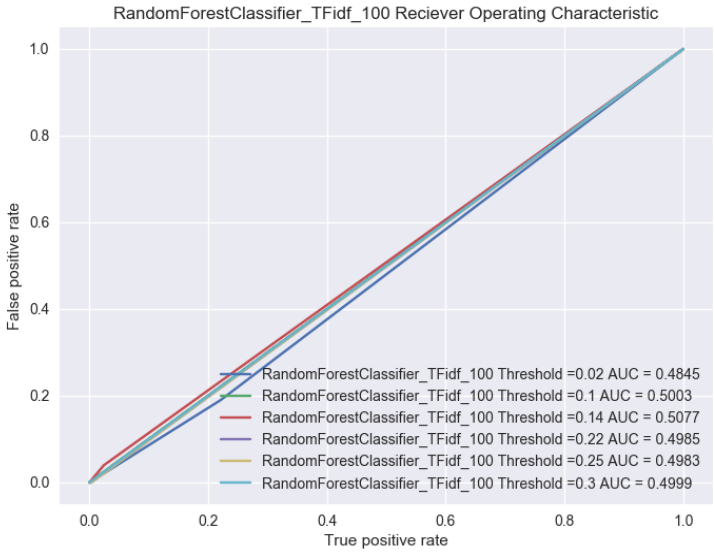

Figure 5 ROC_RandomForest on tfidf

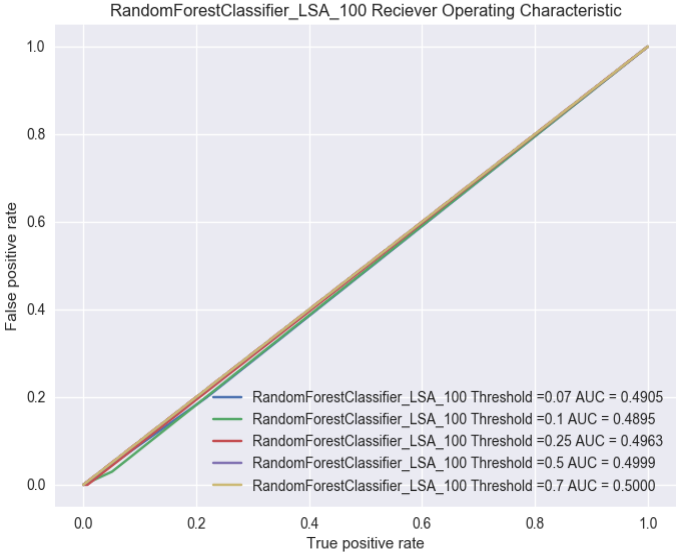

Figure 4 ROC_RandomForest on LSA

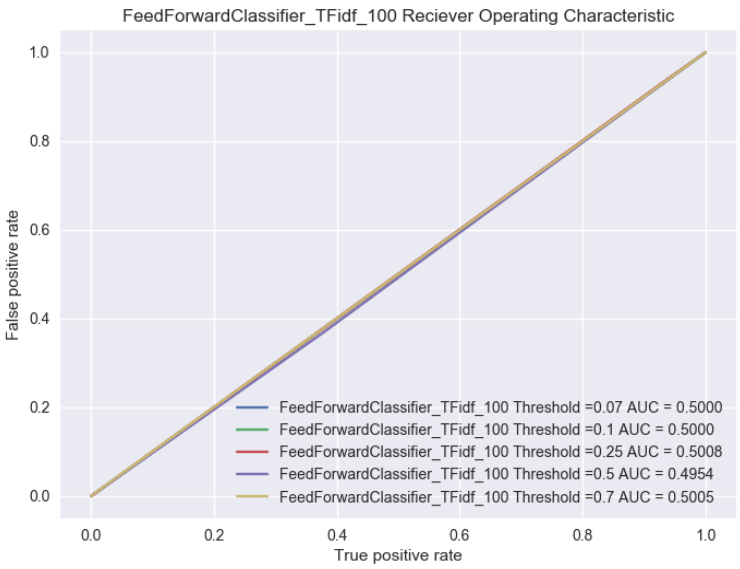

Figure 11 ROC_FeedForward on tfidf

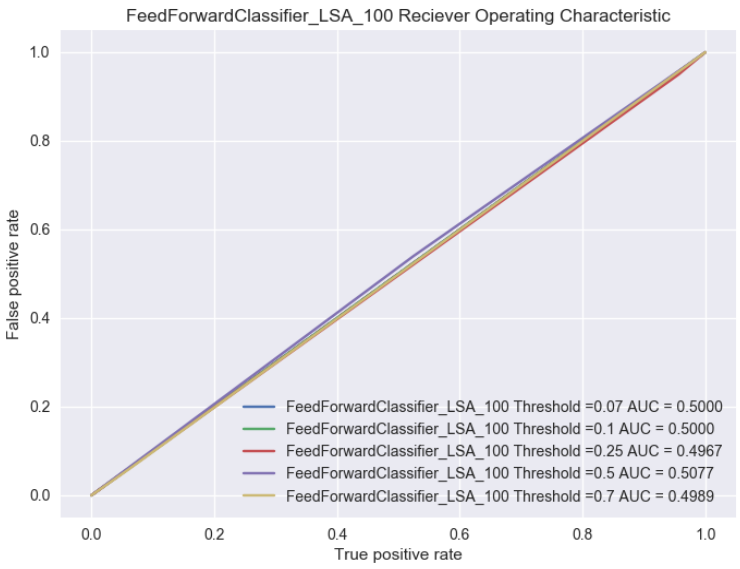

Figure 10 ROC_FeedForward on LSA

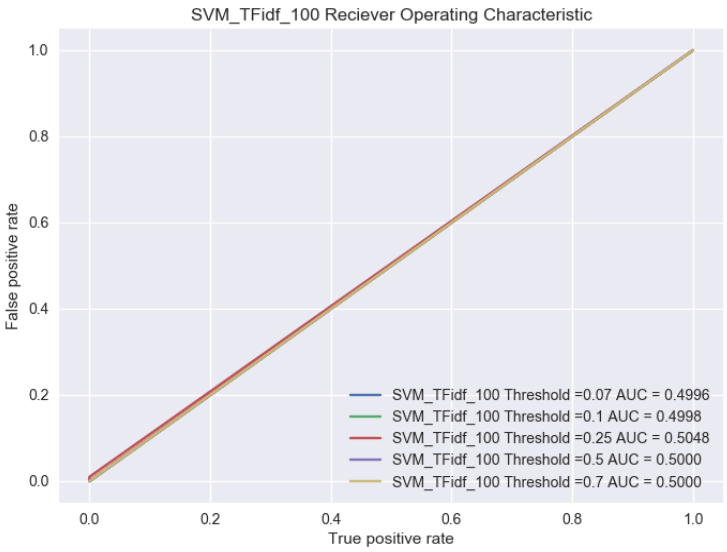

Figure 9 SVM_tfidf_100 Receiver Operating Characteristic

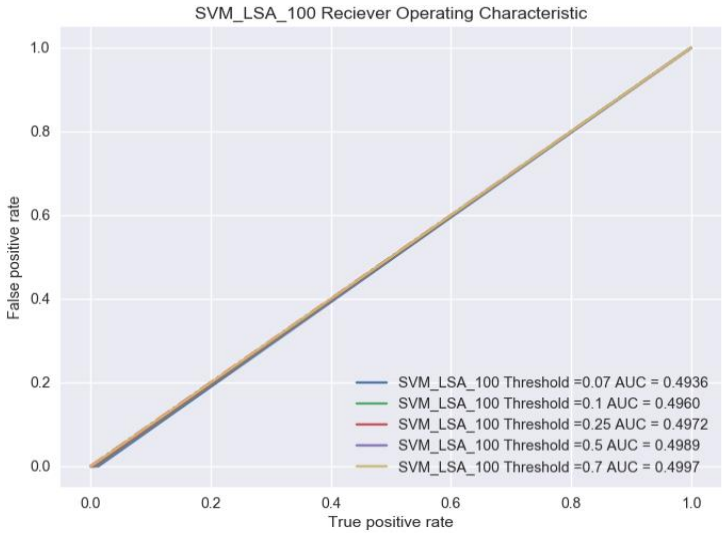

Figure 8 SVM_LSA_100 Receiver Operating Characteristic

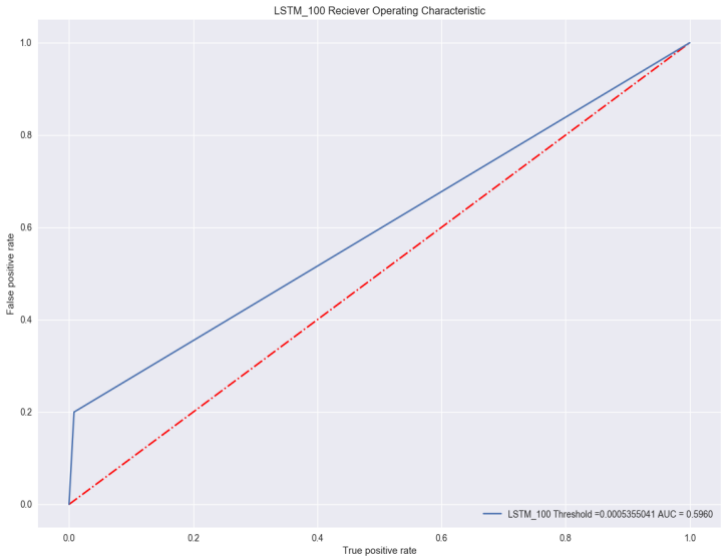

Figure 13 LSTM_100 Receiver Operating Characteristics

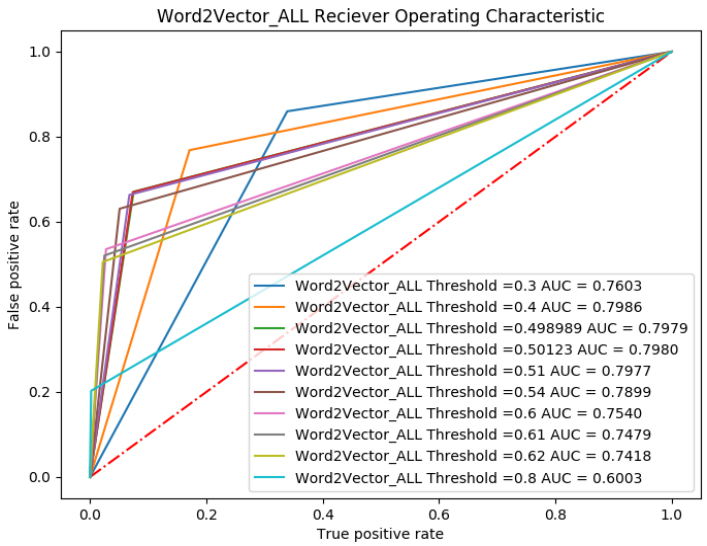

Figure 12 W2V_All Receiver Operating Characteristics

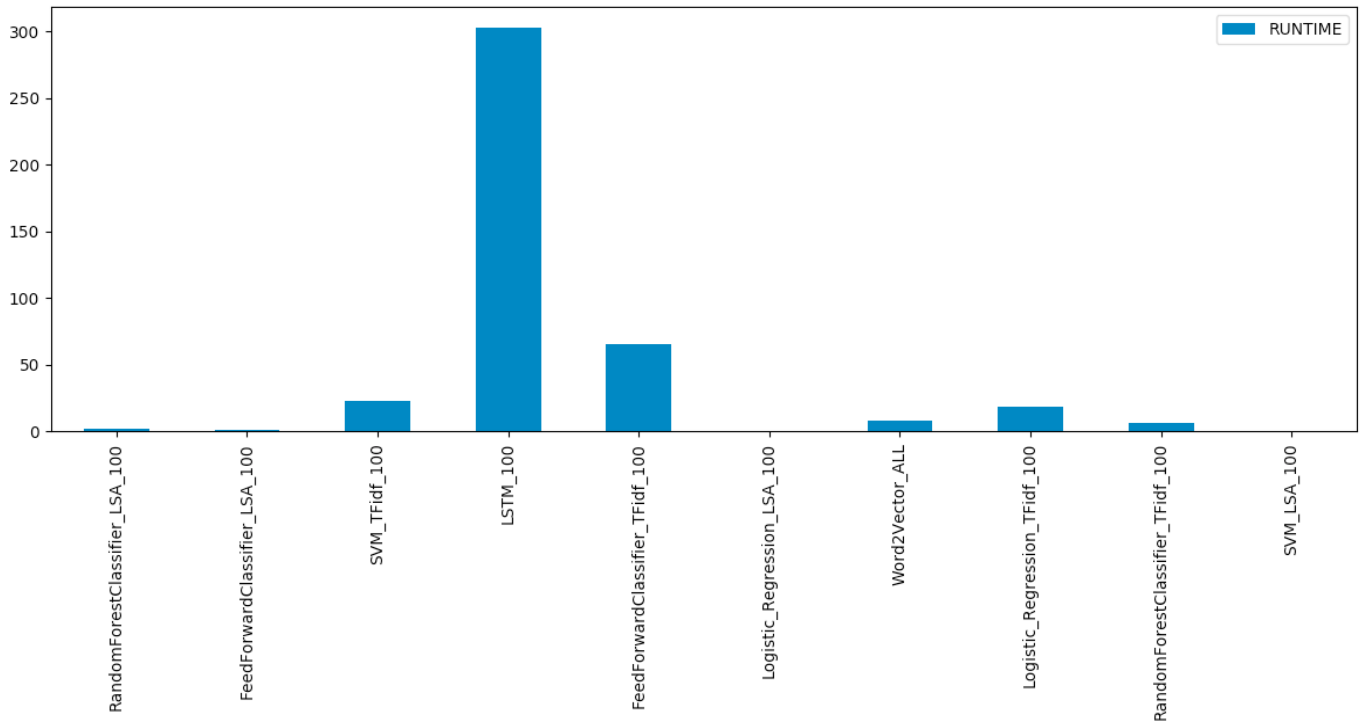

Figure 14 Run Time

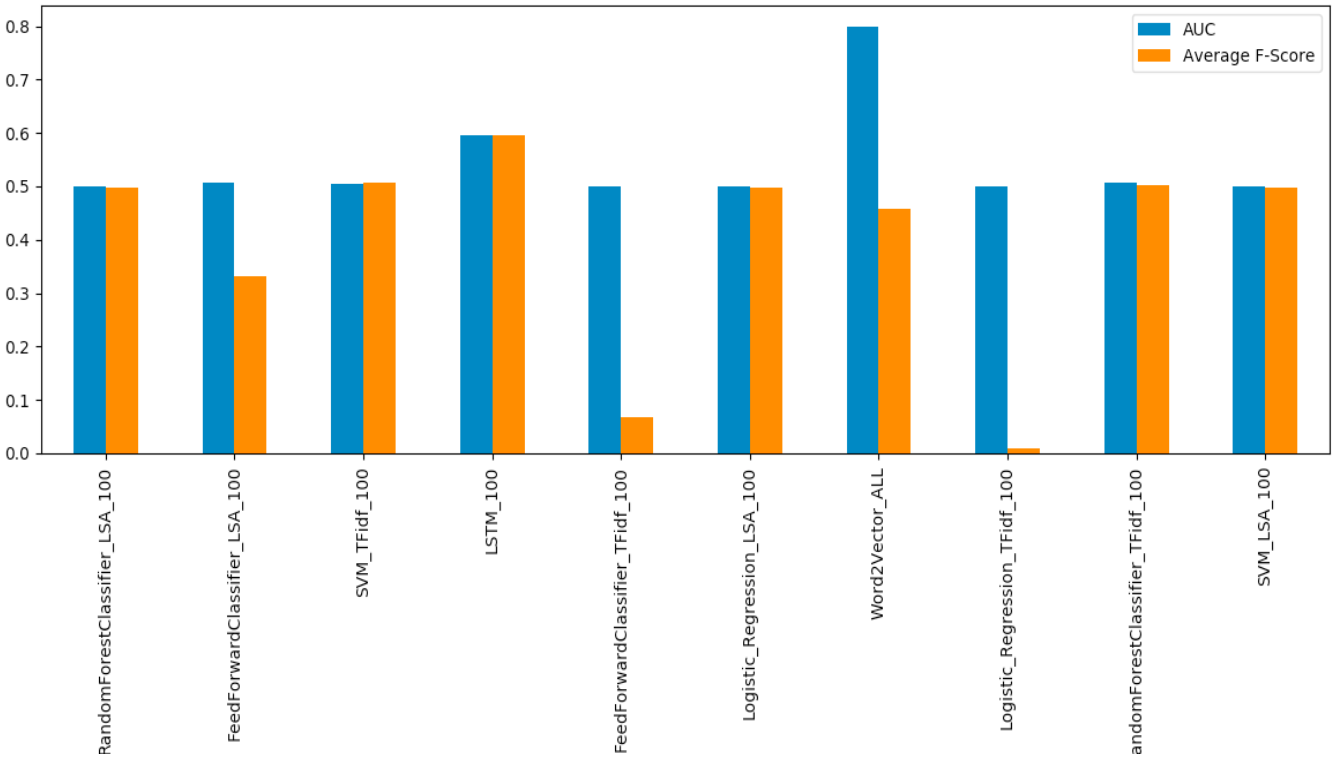

Figure 15 AUC and Average F-Score

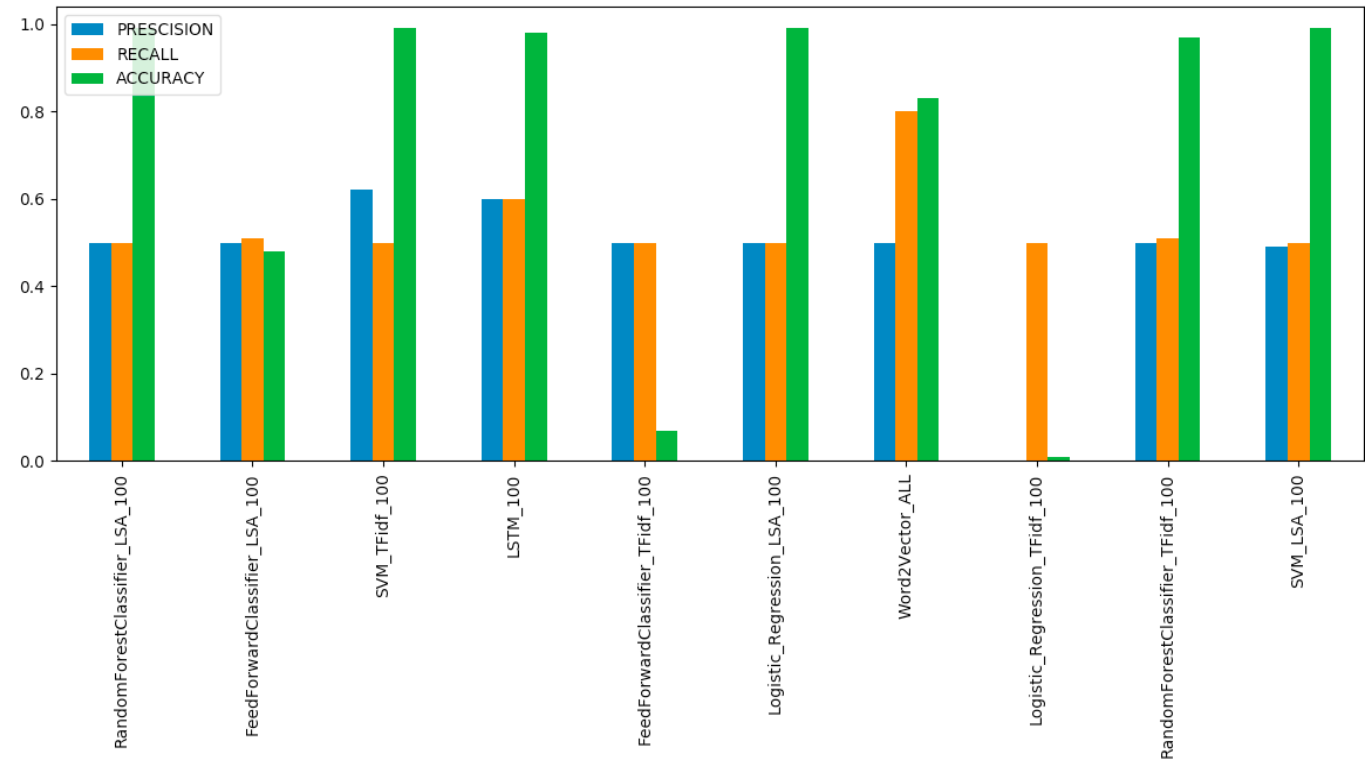

Figure 16 Precision, Accuracy, and Recall

5. CONCLUSION

In this paper, we evaluated different feature extraction methods and NLP deep learning based models, for automatic coding assignment of clinical ICD – 10 codes. The deep learning models for predicting the ICD -10 codes were better than our baseline models that use traditional learning models. The purpose of this paper was to serve as a baseline for future research on automation of ICD coding.

6. FUTURE RESEARCH

To further improve the prediction accuracy, we believe that more advanced networks architectures should be implemented for our data. Although LSTM and Word2Vector can capture long-term dependencies, the length of our input sequence could still be too long for LSTM and Word2Vector to retain useful information. Our current baseline models for top 100 ICD-10 codes and categories were not as successful, may be because the model design lacks capability in effectively distinguishing between 100 different labels, especially for a clinical database where most of the labels are similar to one another. To improve this model, a run of 10 top 10 models in parallel can be performed with each model predicting 10 labels. Further research on what words affect the probability of a prediction could improve our understanding of the relationship between symptoms and diagnosis. The probability observation could also change our preprocessing and feature extraction methods and ultimately improve our deep learning models.

References

[i] Organization, W. H. et al. International classification of diseases:[9th] ninth revision, basic tabulation list with alphabetic index. World Heal. Organ. (1978).

[ii] "International Classification of Diseases (ICD)". World Health Organization. Archived from the original on 12 February 2014.

[iii] Sheppard, J. E., Weidner, L. C., Zakai, S., Fountain-Polley, S. & Williams, J. Ambiguous abbreviations: an audit of abbreviations in paediatric note keeping. Arch. disease childhood 93, 204–206 (2008).

[iv] Lang, D. Consultant report-natural language processing in the health care industry. Cincinnati Child. Hosp. Med. Center, Winter (2007).

[v] Goldstein I, Arzrumtsyan A, Uzuner O. Three approaches to automatic assignment of ICD-9-CM codes to radiology reports. *AMIA Annu Symp Proc*. 2007;2007:279-83. Published 2007.

[vi] Rios A, Kavuluru R. Supervised Extraction of Diagnosis Codes from EMRs: Role of Feature Selection, Data Selection, and Probabilistic Thresholding. *IEEE Int Conf Healthc Inform*. 2013;2013:66-73.

[vii] Farkas R, Szarvas G. Automatic construction of rule-based ICD-9-CM coding systems. *BMC Bioinformatics*. 2008;9 Suppl 3(Suppl 3):S10. Published 2008 Apr 11. doi:10.1186/1471-2105-9-S3-S10

[viii] de Lima LRS, Laender AHF, Ribeiro-Neto BA: **A hierarchical approach to the automatic categorization of medical documents.** In *CIKM '98: Proceedings of the seventh international conference on Information and knowledge management*. New York, NY, USA: ACM Press; 1998:132–139

[ix] Crammer K, Dredze M, Ganchev K, Talukdar PP, Carroll S. Automatic Code Assignment to Medical Text. BioNLP 2007: Biological, translational, and clinical language processing. Prague, CZ; 2007:129-36.

[x] Baud R. A natural language based search engine for ICD10 diagnosis encoding. Med Arh 2004:79-80.a

[xi] Hofmann, T. Probabilistic latent semantic analysis. In Uncertainty In Artificial Intelligence (UAI 1999)

[xii] Blei, David M., Ng, Andrew Y., and Jordan, Michael I. Latent Dirichlet allocation. J. Mach. Learn. Res., 3:993– 1022, March 2003. ISSN 1532-4435.

[xiii] (PDF) Feature Extraction or Feature Selection for Text Classification: A Case Study on Phishing Email Detection. Available from:

https://www.researchgate.net/publication/276431000_Feature_Extraction_or_Feature_Selection_for_Text_Classification_A_Case_Study_on_Phishing_Email_Detection [accessed Dec 02 2018].

[xiv] T. Hofmann, Unsupervised Learning by Probabilistic Latent Semantic Analysis, Mach. Learn. 42 (1-2) (2001) 177–196. URL http://dx.doi.org/10.1023/A:1007617005950

[xv] Min-Ling Zhang and Zhi-Hua Zhou. A review on multi-label learning algorithms. *IEEE transactions on knowl- edge and data engineering*, 26(8):1819–1837, 2014.