Final Project – NLP for Healthcare

Sang Yoon Hwang, MS in Computer Science, Prateek Gupta, MS in Computer Science,
Venkata Shyam K Tumuluri, MS in Computer Science,
Georgia Institute of Technology, Georgia, Atlanta, United States of America

Code: https://github.gatech.edu/pgupta353/CAML_GroupProject/tree/master/src
Presentation: https://www.youtube.com/watch?v=F-pA0zNAn64

Abstract

Clinical text documents such as clinical notes are often annotated by medical codes that uniquely identify each patient's diagnosis, health condition and treatment type. Finding the connection between each clinical note and the codes is important process to understand the reasoning behind diagnoses and why each patient is taking specific treatment. This task often requires manual human effort and therefore prone to errors. A patient's discharge summary can contain multiple medical codes. To minimize human errors and unnecessary efforts, we will present machine learning techniques such as convolutional neural network (CNN), Gated Recurrent Unit Network (GRU) and long short-term memory networks (LSTM) that predict medical codes from clinical text documents. The purpose of this paper is to reproduce the result from the paper "Explainable Prediction of Medical Codes from Clinical Text" by James Mullenbach, Sarah Wiegreffe, Jon Duke, Jimeng Sun and Jacob Eisenstein using CNN, GRU and LSTM algorithms. The results of macro AUC for average value on test sets for each model are following; CAML train/test with 0.973/0.971, GRU train/test with 0.955/0.954 and LSTM train/test with 0.956/0.956. The best performing model on test set was CAML model. The size of the labels recorded for training is 8929 with total number of observations of 42181 in train sets and 5291 in test sets. Top 5 ICD-9 code predictions in terms of probability score ranking for each model on test sets are following; CAML (401.9,427.31,414.01,428.0,38.93), GRU (401.9,38.93,427.31,428.0,414.01) and LSTM (401.9,38.93,427.31,96.04,428.0)

Introduction

Clinical notes/Discharge summaries are produced during patient admissions. The summaries contain unstructured text about diseases, treatments provided to the patient and their response. The ICD-9 taxonomy provides ~13000 unique codes for classifying diseases encountered during patient visits. Manually annotation of discharge summaries is time taking and error prone and each summary can be tagged to multiple ICD-9 codes. Automation of the tagging process was previously tried with ML techniques. Large datasets is a secondary dimension of this problem – Discharge summaries can be large and unstructured and working with discharge summaries could not be done on a single machine. Using CNN and RNN models for solving NLP problems has yielded models with higher accuracy. In this paper, we are going to use the word2vec embedding representation (T. Milov 2017 6) of each word in discharge summary documents as features to 3 models:

- CNN model with Attention mechanism
- GRU
- LSTM

Each model is designed to predict multiple ICD-9 codes using word2vec embeddings. The models are effective in learning specific snippets of text which are effective in predicting an ICD-9 code.

Methodologies

Data Processing and Cleaning

For processing discharge summaries at scale, we are initially loading Mimic III datasets into HDFS (Hadoop 3.2) and process the datasets using Spark 3.1

Data sources

- Discharge summaries corresponding to patient summaries are provided in NOTEEVENTS.csv

- All diagnostic ICD-9 codes and procedural ICD-9 codes are provided in DIAGNOSES_ICD.csv and PROEDURES_ICD.csv files

ETL Processing steps

- The following 3 files - NOTEEVENTS.csv, DIAGNOSES_ICD.csv and PROEDURES_ICD.csv into HDFS from a scalability perspective.
- Raw hdfs files are consumed into Spark as data frames for further processing steps.
- Diagnostic and procedure ICD-9 codes are formatted based on rules for the first 3 or 4 prefixes of the ICD-9 codes - https://www.cms.gov/Medicare/Quality-Initiatives-Patient-Assessment-Instruments/HospitalQualityInits/Downloads/HospitalAppendix_F.pdf
- Filtering clinical notes of patients by type "Discharge Summaries" and restricting it to those notes corresponding to the processed ICD-9 codes provided in the previous step.
- Tokenize the words in each document. Stop words are not removed from the document as we realized that removal of stop words would alter the meaning behind a medical event described in the clinical notes. E.g: Removing the stop word "***not***" in the phrase "Insulin levels are *not* effected" has altered the meaning.
- Generate a list of ICD-9 codes corresponding to clinical notes. A combination of SUBJECT_ID (patient) and HADM_ID (admission id) would provide the list of ICD-9 codes - which can be grouped to form a list.
- Split the dataset into Train, Test and validation datasets and ensure that summaries corresponding to datasets are confined to a single dataset.
- Create a vocabulary for all documents in the training set.

Feature Extraction

WordToVec embedding are created using spark API for every tokenized word from a document. For all 3 models, the embedding size is set to 100. A dictionary would host the mapping. The dictionary maps an integer value to the word embedding. Set the index of –1 to represent padding word2vec index -padding index has zero values in the embedding. Set an index which is one more than the length of vocabulary to represent unknown vocabulary values. The embedding for unknown words would contain random values.

The number of words in each document are limited to 500. Each WordToVec representation is converted into a 3D tensor of the shape BatchSize X MaxLengthofWords X 1, where 1 is the size of the index. This is a sparse representation using indices of the embeddings. The input 3D tensor set being fed into the network would be converted to a tensor of fize BatchSize X MaxLengthOfWords X EmbeddingSize. The embedding size used for the models is 100.

The shape of the labels is a 1D tensor consisting of One hot encoding of each ICD-9 code tagged to corresponding summary.

Model Approach

LSTM

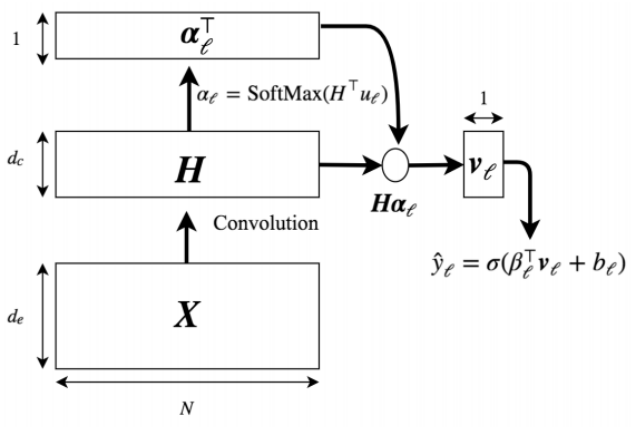
Our approach is consistent with the paper, "Explainable prediction of medical codes from clinical text", with exception of including LSTM model. According to the paper "Regularizing and Optimizing LSTM Language Models" and "Recurrent Neural Network for Text Classification with Multi-Task Learning", we learned that LSTM maybe a good addition for NLP prediction. Long Short-Term Memory (LSTM) networks are a type of RNN which can learn order dependence in sequence prediction problems. While standard RNNs fail to learn when there are time lags greater than 5 to 10, between relevant input events and target signals, LSTM is not affected by this problem. LSTM can learn to connect time lags by enforcing constant error flow through constant error carrousels (CECs) within "cells". Parameter grid we used for LSTM is following:

(embeddings): Embedding(57205, 100)
(LSTMCell): LSTM(100, 32, batch_first=True)
(FullyConnectedOutput): Linear(in_features=32, out_features=8929, bias=True)

Word embeddings collected from Data collection and preprocessing phase would be used for training a CNN with attention mechanism (CAML), GRU and LSTM. The model performance for each model is then compared against each other. We are providing the details of the CNN attention mechanism below as it is the prime model compared against other models.

CNN Attention

Referring to the CAML model provided in the paper, "Explainable prediction of medical codes from clinical text"



Our CAML model has the following calculations in the network assuming a batch size of 32 and maximum number of words allowed in the text is 500 and the embedding size is 100. We have also used 50 as output filter size. The size of the labels recorded for training is 8929.

Input Size: torch.Size([32, 500, 1])
Conversion of the tensor into embedding layer: torch.Size([32, 500, 100])
Application of convolution and tanh activation functon: torch.Size([32, 500, 50])
Attention vectors for each label collected into a matrix: torch.Size([8929, 50])
Alpha , which is the attention layer used instead of maxpooling: torch.Size([32, 8929, 500])
Vector maxtrix when alpha multiplies with convolution output: torch.Size([32, 8929, 50])
Output shape which is a list vectors for producing a binary classification for each label: torch.Size([32, 8929])

GRU

GRU (gated recurrent units) are mechanism in RNN. It is similar to LSTM in terms of architecture having a forget gate but with fewer parpameters.

Parameter grid we used for GRU is following:

(embeddings): Embedding(57205, 100)
(GRUCell): GRU(100, 16, batch_first=True, bidirectional=True)
(FullyConnectedOutput): Linear(in_features=32, out_features=8929, bias=True)

We are using a Binary Cross Entropy with logistic loss as the loss function, which would apply the sigmoid function, providing the probabilities for each ICD-9 code for all 3 models.

Metrics for Measurement

The problem space is a multi-label classification task. The evaluation metric which are used to evaluate the performance of our model are the following:
• Micro average AUC and F1 score which is average of AUC scored across each prediction (text and single label)
• Macro average AUC and F1 score which is average of AUC across each label (to know the performance of rare labels)

Experimental Results

**** bracket = average value**

Micro AUC	Macro AUC	Micro F1	Macro F1	Value Type	Model Type
0.973 (0.974)	0.965 (0.973)	0.428 (0.422)	0.761 (0.744)	Train	CAML
0.967 (0.973)	0.974 (0.971)	0.387 (0.430)	0.834 (0.750)	Test	CAML
0.951 (0.957)	0.942 (0.955)	0.190 (0.226)	0.481 (0.547)	Train	GRU
0.958 (0.957)	0.958 (0.954)	0.200 (0.238)	0.589 (0.551)	Test	GRU
0.963 (0.960)	0.955 (0.956)	0.260 (0.235)	0.571 (0.548)	Train	LSTM
0.954 (0.960)	0.871 (0.956)	0.278 (0.248)	0.631 (0.546)	Test	LSTM

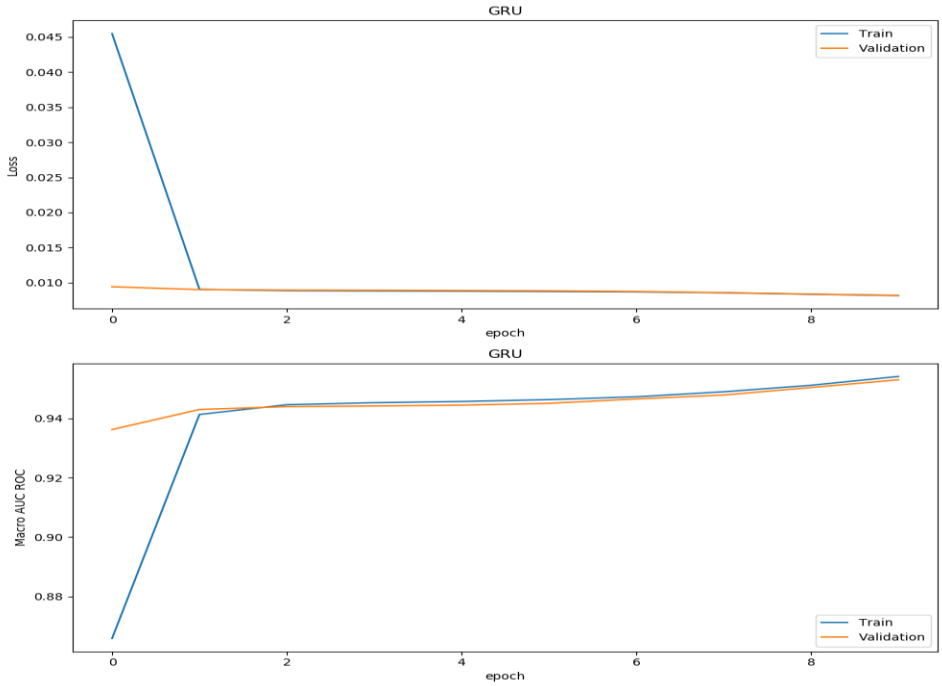
Parameters grid used to run models

batch size = 32, epoch=10, optimizer = optim.Adam(model.parameters(), lr=1e-4) with BCEwithLogitsLoss

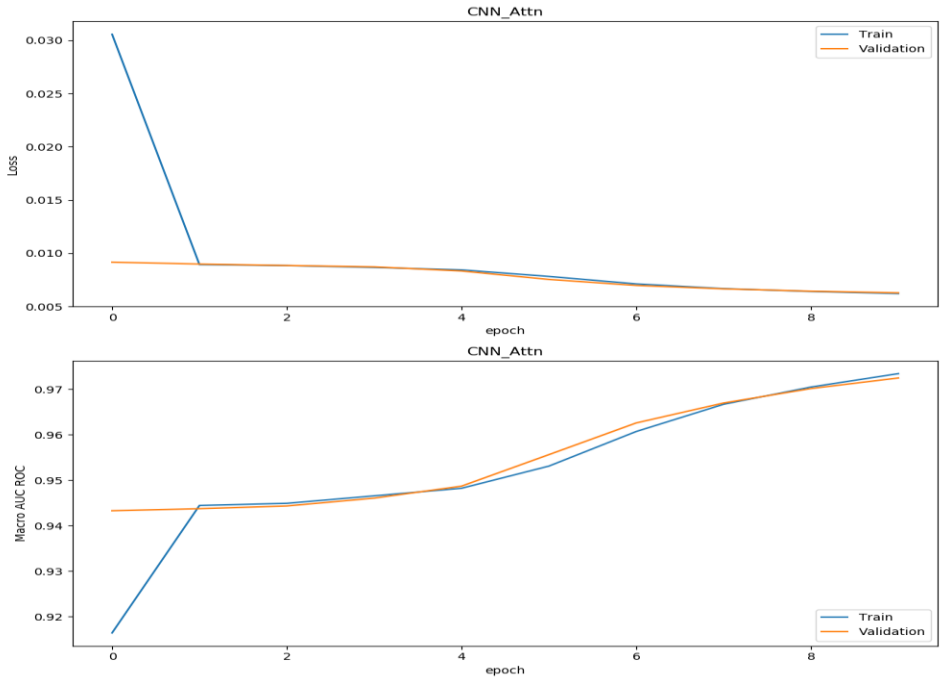
Discussion

- From the experimental results, we noticed that CAML was the best performer in terms of both AUC and F1, notably with macro AUC with 0.972 for average value on test set.
- From the loss graph below, we confirmed that loss for each model is becoming smaller as epoch increases. To prevent overfitting issue, we implemented Xavier initialisation.
- Top 5 ICD-9 code predictions in terms of probability score ranking for each model on test sets are following; CAML (401.9,427.31,414.01,428.0,38.93), GRU (401.9,38.93,427.31,428.0,414.01) and LSTM (401.9,38.93,427.31,96.04,428.0)

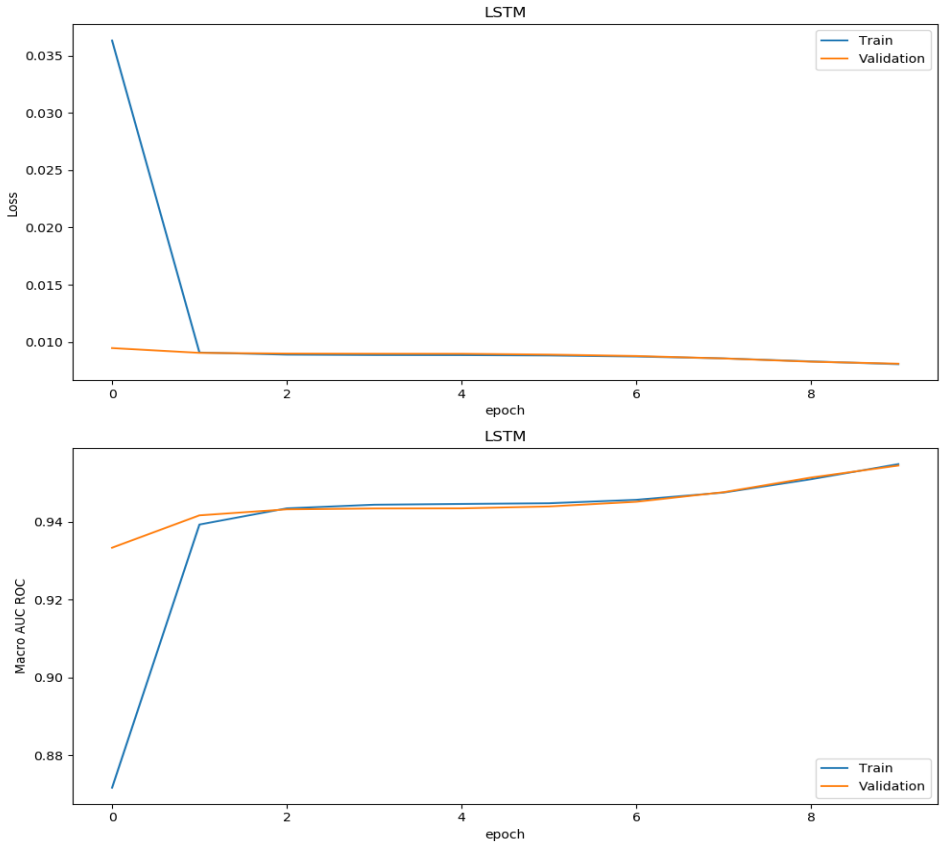
GRU



CAML


CNN_Attn
Train
Validation
Loss
epoch
CNN_Attn
Macro AUC ROC
Train
Validation
epoch

LSTM


LSTM
Train
Validation
Loss
epoch
LSTM
Macro AUC ROC
Train
Validation
epoch

Conclusion

For this project, we used multi-label classification methods, CAML, GRU and LSTM, with trained word vectors from the clinical notes to predict ICD-9 codes. We then configured layers and corresponding parameters for each model to understand where models produce the highest F1 and AUC. To handle big data pre-processing, we used big data technology such as PySpark. Furthermore, to counter overfitting issue, we implemented Xavier initialisation and the best performed model was CAML – notably, with macro AUC for average value of 0.971 on test set.

References

1. J. D. J. S. J. E. James Mullenbach, Sarah Wiegreffe. Explainable prediction of medical codes from clinical text. ACL Anthology, 2018.
2. T. Kang, S. Zhang, Y. Tang, G. W. Hruby, A. Rusanov, N. Elhadad, and C. Weng. Eliie: an open-source information extraction system for clinical trial eligibility criteria. Journal of the American Medical Informatics Association, 24(6):1062–1071, 2017.
3. Z. He, S. Carini, I. Sim, and C. Weng. Visual aggregate analysis of eligibility features of clinical trials. Journal of biomedical informatics, 54:241–255, 2015
4. H. Ma and C. Weng. Identification of questionable exclusion criteria in mental disorder clinical trials using a medical encyclopedia. In Biocomputing 2016: Proceedings of the Pacific Symposium, pages 219– 230. World Scientific, 2016.
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, 60:132–144, 2016.
6. Toma s Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch and Armand Jouli. Advances in pre-training distributed word representations. CoRR, abs-1712-09405, 2017.
7. Stephen Merity, Nitish Shirish Keskar, Richard Socher. Regularizing and Optimizing LSTM Language Models. arXiv:1708.02182, 2017.
8. Pengfei Liu, Xipeng Qiu and Xuanjing Huang. Recurrent Neural Network for Text Classification with Multi-Task Learning. arXiv:1605.05101, 2016.