Final Project Report for CS598 DL4H in Spring 2023

Aareana Reza and Brittany West
{areza5, bnwest2}@illinois.edu

Group ID: 20
Paper ID: 162
Presentation link: https://youtu.be/oMtv60hFCew
Code link: https://github.com/AareanaReza/CS598-DLH-Final-Project

1 Introduction

Electronic Health Records (EHR) are complex and lengthy histories of a patients well being and are often used in neural networks to predict future patient outcomes. Because of EHR's complexity there is a lot of room for exploration, in particular when it comes to the free text data that the EHR carries. The paper "Real-world Patient Trajectory Prediction from Clinical Notes Using Artificial Neural Networks and UMLS-Based Extraction of Concepts" aims to predict better healthcare related outcomes for patients by utilizing EHR's clinical notes which are unstructured text. They preprocessed the text, extracted concepts, and used fine-tuning neural networks to achieve this. For our project, we will be doing a reproduction study of this paper and comparing the results.

2 Scope of reproducibility

2.1 Addressed claims from the original paper

- Claim 1: *Through their experimentation with different models, the researchers found that using the Feed-Forward architecture resulted in more accurate results on this dataset than when Recurrent Neural Networks were used.*

  The Recurrent Neural Network uses the entire history of admissions of a patient while Feed-Forward architecture only relies on the last admission of a patient. It is interesting that having more history of a patient might have negative effects when predicting their future, which is why we have chosen to investigate this claim further.

3 Methodology

3.1 Model descriptions

For our reproduction study, we focused on the models used for this paper's diagnosis prediction model. One of the architectures used was the fully connected Feed-Forward architecture. This architecture only takes into account the last admission in its calculations. Another architecture they used was a Gated Recurrent Unit which is a type of Recurrent Neural Network. RNN takes into account all of a patient's previous admissions in its algorithm.

Other parameters they focused on were selecting the best threshold and list of Type Unique Identifiers (TUI). These parameters then generated the best set of Concept Unique Identifier (CUI) codes which were used as inputs to improve the performance of the diagnosis prediction task.

One of the objectives of this paper is to determine which combination of architectures and parameters is the best way to predict future patient diagnosis with clinical notes as input.

3.2 Data descriptions

The original study uses MIT's MIMIC-III dataset, which contains de-identified medical information about real patients. We were able to use this database by requesting access for it through PhysioNet. Specifically, we used the `ADMISSIONS.csv`, `DIAGNOSES_ICD.csv`, and `NOTEEVENTS.csv` files from the dataset. The `NOTEEVENTS.csv` required some preprocessing before it was ready to be used in the deep learning predictive model. After following the necessary data cleaning steps that were documented in the researcher's code, we were able to easily review the data further. We have removed rows that have outlier document length, based on a 5% cutoff. We have also removed patients that have less than 10 and more than 20 clinical documents in order to reduce the amount of patients we are sampling from. The clinical notes

originally contained information about 46,146 patients and had 2,083,179 documents that were on average 1826 lines long. After doing data cleaning we were left with 9,857 patients. We then took a 1% sample of this data and used information on 99 patients from 2,546 documents for our analysis.

This study also uses a list of CUI codes that are obtained from using QuickUMLS. QuickUMLS is a tool for concept extraction and requires a license from the National Library of Medicine in order to use it.

3.3 Hyperparameters

The researchers adjusted different hyperparameters when creating their predictive model such as number of epochs, learning rate, batch size, optimizers,number of hidden layers, and dropout. They reported the best results when they set the number of epochs to 5,000 for FFN and 1,500 for RNN, the batch size to 100 for FFN and 10 for RNN, and used optimizers Stochastic Gradient Descent for FFN and Adam for RNN.

3.4 Implementation

We are referencing the researcher's code to perform our analysis and reproduction of their diagnosis prediction model.
Link to researcher's code: https://github.com/JamilProg/patient_trajectory_prediction
Link to our reproduction code: https://github.com/AareanaReza/CS598-DLH-Final-Project

3.5 Computational requirements

The researchers used hardware NVIDIA Quadro P6000 GPU to perform their prediction task. We are using Google Colab Pro in order to be able to replicate their study. The type of hardware that Google Colab Pro uses comes with Intel Xeon CPU @2.20 GHz, 13 GB RAM, Tesla K80 accelerator, and 12 GB GDDR5 VRAM. The average runtime for each epoch was about 1 second and the number of training epochs was 5,000.

4 Results

For our reproduction study we have performed data cleaning on the clinical notes and did concept extraction to get the diagnosis codes. The clinical notes were very large so it took a few days to obtain and process them so that they were suitable for use in the prediction model. We worked with a smaller subset of the clinical notes data, since preprocessing for that file required more computational power and memory then we currently had access to. This means that our results differ from those shown in the original paper. Our final results are detailed below.

4.1 Result 1

Original Paper's Results

	Original Paper's FFN Results	Original Paper's RNN Results
Precision@1	0.75	0.54
Precision@2	0.688	0.508
Precision@3	0.638	0.479
Recall@10	0.392	0.3
Recall@20	0.576	0.458
Recall@30	0.689	0.571
AUC-ROC	0.913	0.872

Our Results

	Our FFN Results	Our RNN Results
Precision@1	0.27	0.205
Precision@2	0.3	0.071
Precision@3	0.29	0.095
Recall@10	0.258	0.285
Recall@20	0.375	0.448
Recall@30	0.481	0.571
AUC-ROC	0.658	0.7

We have run the researcher's base models with the default hyperparameters that they used. Our calculations are overall smaller than those shown

in the original paper. We were able to get the results shown above which shows that the Feed Forward Model performs better than the RNN when it comes to measuring precision. Precision is the measurement of positive identifications that were correctly classified. On average, the original paper's precision was 0.18 higher for FFN than RNN. And, on average, our precision for FFN was 0.16 higher than RNNs precision.

We were not able to recreate the same results when it came to recall. Recall is the number of actual positive results that were correctly identified. On average, the original paper's recall was 0.11 higher for FFN than RNN. We found that on average, the recall for RNN was 0.06 better than FFNs. Our AUC-ROC score was also better for RNN than FFN. The AUC-ROC score refers to the efficiency of the model and its ability to distinguish between positive and negative results. Overall, we were only able to partially recreate the results in this study.

4.2 Ablation Experiment Results

FFN Results with Different Hyperparameters

	Our FFN Results with their hyperparamters	Our FFN Results with our hyperparameters
hiddenDimSize	10,000	50
batchSize	100	1,000
nEpochs	5,000	10,000
Precision@1	0.27	0.43
Precision@2	0.3	0.31
Precision@3	0.29	0.27
Recall@10	0.258	0.238
Recall@20	0.375	0.368
Recall@30	0.481	0.491
AUC-ROC	0.658	0.665

We decided to change the hiddenDimSize parameter to 50 from 10,000 because the input layer we are using in the model is dramatically smaller than the input layer used in the research paper. We made the batchSize and nEpochs parameters bigger going from 100 batches to 1,000 and 5,000 to 10,000 epochs so that training time would increase. By doing these ablations we noticed an increase in precision compared to our original results using the hyperparameters from the paper. There was also a .01 increase to the AUC-ROC score. These improvements are likely attributed to changing the hyperparameters to better fit the smaller amount of data we were able to train.

5 Discussion

5.1 What was easy

The original paper was well-documented and had very clear instructions outlined in the researcher's code and README file. All the necessary steps were very clear and detailed which allowed us to perform analysis without having to worry about preprocessing the files from scratch. This made going through their experiments, understanding their findings, and reproducing their study much more manageable.

5.2 What was difficult

We found it very difficult to do the data preprocessing steps on the input files because it required a lot of computation effort to extract the large amounts of data. We initially attempted to do it locally and through Google Colab, both of which errored out when trying to process the files. We then used Google Colab Pro. It was still not powerful enough to handle the input files and kept timing out during the preprocessing steps. This is why we chose to only take a subset of the data to perform analysis for our project. This made our calculations vastly different from the original results. Overall, all our calculations had smaller values and our recall and AUC-ROC scores showed that RNN performed better than FFN, which opposed the researcher's findings. If we were able to perform the model on the full dataset, we might have been able to recreate the original results more accurately.

Another obstacle we faced was using QuickUMLS. Installing QuickUMLS took a long time because of the large amount of data. Since it required such a big effort, we were only able to conceptualize 2,000 lines out of 2 million. This most likely also contributed to the differences between our results and the researcher's results.

Since it took some time to find the best way to process the input files, we were only able to

recreate the diagnosis prediction experiment in this study and chose to skip the readmission and mortality predictions. We also were unable to test the researcher's second claim that adding diagnosis codes to the input of the prediction model increased accuracy of the results.

5.3 Recommendations for reproducibility

One thing we would recommend to the authors to mention in their README file is that installing QuickUMLS requires a license from the National Library of Medicine. This would have been a useful disclaimer to have when we were initially reading through their project and trying to understand all the dependencies that were required to run their code.

6 Communication with original authors

We were not able to communicate with the original authors about our reproduction study.

References

1. Zaghir J, Rodrigues-Jr JF, Goeuriot L, AmerYahia S. Real-world Patient Trajectory Prediction from Clinical Notes Using Artificial Neural Networks and UMLS-Based Extraction of Concepts. J Healthc Inform Res. 2021 Jun 5;5(4):474-496. doi: 10.1007/s41666-021-00100-z. PMID: 35419508; PMCID: PMC8982755.
   https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8982755/#CR22

2. Original Paper's github link.
   https://github.com/JamilProg/patient_trajectory_prediction

3. Our reproduction code github link.
   https://github.com/AareanaReza/CS598-DLH-Final-Project

4. MIT's MIMIC-III Clinical Database.
   https://physionet.org/content/mimiciii/1.4/#

5. Unified medical language system (UMLS).
   https://www.nlm.nih.gov/research/umls/index.html

6. Quickumls github link.
   https://github.com/Georgetown-IR-Lab/QuickUMLS

7. Ford Elizabeth, Curlewis Keegan, Squires Emma, Griffiths Lucy J., Stewart Robert, Jones Kerina H.The Potential of Research Drawing on Clinical Free Text to Bring Benefits to Patients in the United Kingdom: A Systematic Review of the Literature. Frontiers in Digital Health
   https://www.frontiersin.org/articles/10.3389/fdgth.2021.606599

8. Aniruddha Bhandari. Guide to AUC ROC Curve in Machine Learning : What Is Specificity?
   https://www.analyticsvidhya.com/blog/2020/06/auc-roc-curve-machine-learning/

9. Kizito Nyuytiymbiy. Parameters and Hyperparameters in Machine Learning and Deep Learning.
   https://towardsdatascience.com/parameters-and-hyperparameters-aa609601a9ac#:~:text=Hyperparameters%20are20parameters%20whose%20values,parameters%20that%20result%20from%20it