

Reproducibility Project for CS598 DL4H in Spring 2022

Liangwen Wu and Matt McCarty

Group ID: 41, Paper ID: 111

Code link: https://github.com/mdmuiuc/patient_trajectory_prediction

1 Introduction

The usage of Electronic Health Records in the medical profession have increased in usage globally. Part of these Electronic Health Records include clinical notes, mainly comprised of free-form unstructured text. Modern developments in natural language processing and machine learning unlock the potential for such unstructured clinical notes to be used to predict patient outcomes in the future. In this paper, we will attempt to re-create the results of an existing paper (Zaghir et al., 2021) which involve using the Unified Medical Language System (Bodenreider, 2004) to extract medical concepts from clinical notes and a neural network model that outputs a set of probabilities associated with various patient outcomes. The output of the model will be evaluated with metrics such as precision and compared with the performance of other state-of-the-art models such as doc2vec (Le and Mikolov, 2014) and stacked denoising autoencoder networks (Vincent et al., 2010).

2 Scope of reproducibility

The claim we will attempt to reproduce is that the fully connected feed-forward neural network based on the UMLS-derived CUI representation of clinical notes has a higher precision for predicting the primary patient diagnostic outcome than six state-of-the-art models (Sushil et al., 2018):

- BoW (Bag-of-words)
- doc2vec
- SDAE (Stacked denoising autoencoder)-BoW
- [doc2vec, SDAE-BoW]
- BoCUI (UMLS-derived CUI representation+CLAMP) (Soysal et al., 2017)

- SDAE-BoCUI

The specific precision measure, Precision@1, is defined as follows:

$$\text{Precision@k} = \frac{\# \text{ correct diagnostic codes in the top-k}}{k}$$

3 Methodology

The authors provided a GitHub repository (Zaghir et al., Accessed March 2022.) complete with Python scripts to pre-process the data, train the neural network model and compute the evaluation metrics. We will also use QuickUMLS (Soldaini and Goharian, Accessed March 2022.) to extract CUIs from the clinical notes.

3.1 Model descriptions

Apart from the data pre-processing step which is described in the following section, the model is comprised of two main stages. The first stage involves extracting clinical concepts from the free-form notes using a UMLS concept recognizer (Bodenreider, 2004). This concept recognizer evaluates concepts and text by computing a score between 0 (no match) and 1 (perfect match). Based on a threshold value, the cataloged concepts which have a score greater than the threshold are assigned. The concepts are then reduced by a list of Type Unique Identifiers. The paper uses two TUI lists, α and β . These lists are included in the GitHub repository.

After the concept extraction stage is complete, the concepts are represented as one-hot vectors and passed to different network architectures: a fully connected feed-forward neural network and a recurrent neural network. This paper will focus on the first of the two neural network models. The feed-forward neural network model has an input layer, one fully connected hidden layer, and one output layer with 269 nodes for different diagnoses,

1 for readmission and 3 for mortality. A sigmoid activation layer is used to force the output to a probability value in the range (0, 1). The loss function is a binary cross-entropy loss function:

$$\text{Loss} = \frac{1}{N} \sum_{i=1}^N y_i * \log(p(y_i)) + (1 - y_i) * \log(1 - p(y_i))$$

The optimizer used for training is stochastic gradient descent.

3.2 Data descriptions

This paper uses the MIMIC-III dataset (Johnson et al., 2016), specifically the NOTEVENTS.csv file. Some pre-processing is performed. Specifically, stop-words are removed, the text is converted to lowercase, and special/noisy/whitespace characters were removed. In addition, the free-text was segmented to paragraphs, which is essential to the UMLS concept extraction phase.

After performing the data pre-processing and filtering, the data set contained 71,349,141 observations, each containing a string of free text for the clinician’s notes.

3.3 Hyperparameters

There are a number of hyperparameters used throughout the entire model training process.

3.3.1 QuickUMLS Hyperparameters

For QuickUMLS, there are a number of hyperparameters.

Hyperparameter	Description	Value
threshold	The minimum similarity value between strings	0.9
TUI list filter	The TUI list filter	β
window	The maximum number of tokens to be considered for matching	5

3.3.2 Neural Network Hyperparameters

The neural network also has a number of hyperparameters.

Hyperparameter	Description	Value
dropout rate	The dropout rate applied at the end of the neural network	0.5
learning rate	The learning rate of the stochastic gradient descent optimizer	0.01
batch size	The batch size for neural network training	100
epoch size	The number of training iterations	5000
hidden dim size	The number of nodes in the hidden neural network layer	10000

3.4 Implementation

As previously described, we will be using the authors’ codebase (Zaghir et al., Accessed March 2022.) to train the neural network and evaluate it on the MIMIC-III clinical notes. In addition, the QuickUMLS library (Soldaini and Goharian, Accessed March 2022.) will be used for the concept recognizer phase.

Our fork of the codebase, which includes a table of results, can be found at https://github.com/mdmuiuc/patient_trajectory_prediction.

3.5 Computational requirements

3.5.1 Prediction and Planning

The README.md file in the authors’ codebase (Zaghir et al., Accessed March 2022.) mentions that they used a single GPU Quadro P6000 and that the most time-consuming step is the CUI recognizer with QuickUMLS (Soldaini and Goharian, Accessed March 2022.). That step took the authors several hours to 3 days to finish, depending on the specified hyperparameters (the QuickUMLS threshold and the TUI list filter). We have an Nvidia GTX 1050-Ti GPU, which may not be as powerful as a Quadro P6000, but it was able to perform the CUI recognizer stage of the model training pipeline in 18 hours, given the hyperparameters specified above. The data pre-processing step took eight hours to perform.

3.5.2 Actual Results

The most time-consuming step was not the CUI recognizer step (which took 18 hours on a Nvidia GTX 1050-Ti GPU), but the feed-forward neural network. The feed-forward neural network took 60 hours to train. In addition, we ran into issues with

GPU memory: the GPU the researchers used had 24GB, and our GPU only had 4GB, which caused occasional out of memory exceptions when training the neural network model. In the end, we had to use a more powerful GPU than planned for the neural network training (NVIDIA GeForce RTX 2060) which had 12GB of memory.

GPU used (non-neural network): Nvidia GTX 1050-Ti GPU

GPU used (neural network): Nvidia GeForce RTX 2060

GPU Hours per training step:

- Pre-processing NOTEEVENTS.csv: 8 hours
- CUI Recognizer via QuickUMLS: 18 hours
- Data prep for neural network: 1 hour
- Feed-forward neural network: 60 hours
 - Average runtime per epoch: 8.6 seconds
 - Number of epochs per fold: 5,000
 - Number of folds: 5

4 Results

As stated before, the main hypothesis being tested is that the model described above has a higher precision in predicting diagnostic outcomes than six baseline models specified in (Sushil et al., 2018). The P@1 measure of our trained model is included with the P@1 from the original paper, as well as the six baseline models, below:

Architecture	P@1
FFN (CUI + CCS), original	0.778
FFN (CUI + CCS), reproduction	0.750
BoW	0.701
SDAE-BoW	0.650
doc2vec	0.681
[doc2vec, SDAE-BoW]	0.679
BoCUI	0.710
SDAE-BoCUI	0.665

The hypothesis that the feed-forward neural network has a higher P@1 measure than other models for patient outcome prediction is confirmed. As shown above, our trained neural network model has a higher precision (0.750) than each of the six baseline models. That being said, our trained neural network model was less accurate than the original paper.

4.1 Ablation

In addition, we trained a version of the feed-forward neural network with 1/10 the hidden states of the original neural network (i.e., 1,000 instead of 10,000). The weaker neural network model turned out to be as performant as the original neural network: the P@1 measure of the ablated model was 0.739 instead of 0.750. Having ten times as many hidden states in the original neural network did not significantly increase the model's performance.

5 Discussion

As outlined above, the original hypothesis of the paper was confirmed, and the feed-forward neural network had a higher precision in predicting patient outcomes than prior state-of-the-art models. The implications of this result are quite important: if negative patient outcomes can be accurately predicted in advance, clinicians would have the opportunity to intervene to improve the health outcome of their patients. Machine-aided diagnoses have the potential to have a strong positive impact in the healthcare industry.

5.1 What was easy

The authors of (Zaghir et al., 2021) did an excellent job in ensuring the reproducibility of their results. Not only did they create a public GitHub repository (Zaghir et al., Accessed March 2022.) with all of the python scripts used to train the model, the README file contained thorough, step-by-step instructions on how to:

- Pre-process the data
- Run QuickUMLS to recognize concepts from the clinical notes
- Perform pre-neural network training data cleaning
- Train the feed-forward neural network

In addition, the original paper clearly described the details of the feed-forward neural network and specified the values of the hyperparameters.

5.2 What was difficult

The difficulties we encountered mainly dealt with acquiring access not only to the MIMIC-III data set, but the UMLS metathesaurus, which we did not realize required approval from the National Library

of Medicine. Had we known that, the metathesaurus access requests would have been submitted well in advance to give us time to perform more experimentation with the neural network hyperparameters and further ablations.

In addition, the authors warned about the time required to perform concept recognition with Quick-UMLS, but there were no time estimates provided for the main feed-forward neural network model training. This stage of the model pipeline was the most time-consuming, by far. In addition, the memory requirements of the neural network model training required a very powerful GPU.

5.3 Recommendations for reproducibility

For other researchers attempting to reproduce the results in (Zaghir et al., 2021), we would recommend acquiring a system with a powerful GPU with greater than 12GB memory.

6 Communication with original authors

We did not need to contact the original authors, as their documentation and codebase were thorough and self-explanatory.

References

- O. Bodenreider. 2004. [The unified medical language system \(umls\): integrating biomedical terminology](#). *Nucleic Acids Res.*, 32(Database issue):D267–D270.
- A. Johnson, T. Pollard, and R. Mark. 2016. [Mimic-iii clinical database \(version 1.4\)](#). *PhysioNet*.
- Q. Le and T. Mikolov. 2014. Distributed representations of sentences and documents. *International conference on machine learning (PMLR)*, pages 1188–1196.
- Soldaini and Goharian. Accessed March 2022. [Quick-umls codebase](#).
- E. Soysal, J. Wang, M. Jiang, Y. Wu, S. Pakhomov, H. Liu, and H. Xu. 2017. [Clamp — a toolkit for efficiently building customized clinical natural language processing pipeline](#). *Journal of the American Medical Informatics Association*, 25(3):331–336.
- M Sushil, S Suster, K Luyckx, and W Daelemans. 2018. Patient representation learning and interpretable evaluation using clinical notes. *J. Biomed. Informatics*, pages 84–103.
- P Vincent, H Larochelle, I Lajoie, Y Bengio, PA Manzagol, and Bottou L. 2010. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *The Journal of Machine Learning Research*, 11:3371–3408.
- J. Zaghir, J.F. Rodrigues-Jr, and L. et al Goeuriot. 2021. [Real-world patient trajectory prediction from clinical notes using artificial neural networks and umls-based extraction of concepts](#). *J Healthc Inform Res*, 5:474–496.
- J. Zaghir, J.F. Rodrigues-Jr, and L. et al Goeuriot. Accessed March 2022. [Codebase for real-world patient trajectory prediction from clinical notes using artificial neural networks and umls-based extraction of concepts](#).