AI-Driven Clinical Decision Support: Enhancing Disease Diagnosis Exploiting Patients Similarity

Amit Bhagwat and Rajan Patel
amit4@illinois.edu, rajansp2@illinois.edu

Group ID: 74 Paper ID: X, Difficulty: easy
Presentation link: https://www.youtube.com/channel/UCIGGb11ZykT5mGHZTCozquw/videos
Code link: https://github.com/amitvb/DiseaseDiagnosis

1 Introduction

This paper is about producing a Clinical Decision Support (CDS) system capable of diagnosing and predicting diseases by exploiting similarities amongst patients. There are several papers in the field which provide a clinical support system but, only capable of identifying a single patient and a single medical condition at a time. This paper, making use of Natural Language Processing and Neural Network exploring the features in Electronic Health Records, provides a novel method to tackle the limitations in other papers as discussed in the original paper. Patient similarity is defined as the similarity between patients' diagnosis and symptoms rather than the dichotomy of a disease being present/absent in a patient[1]. This paper applies sentence level word embeddings based on semantic extracted from medical corpus.

2 Scope of reproducibility

The paper claims that with the combination of supervised and unsupervised learning methods, CDS system can help better identify diseases by combining several heterogeneous health data of different patients and different medical conditions. Specifically the model exploits semantic meaning using Natural Language Processing (NLP) concepts enabling exact diagnosing prediction. Word embedding model can be used to learn patient semantics from data collected in form of EHRs and medical data upon hospitalization. This embedding model in turn can be used to establish patient similarity and disease prediction.

"The proposed CDS system is able to integrate heterogeneous data from various sources, implementing AI and NLP algorithms to aid doctors in predicting diagnosing."

"Sentence embedding models like BioSent2Vec model can be used to identify similarities between patients' symptoms (and any other information like preliminary diagnosis, clinical notes, etc.) to predict diagnoses."

"Increasing the size of dataset will increase the precision and recall as well as increasing value of top-K increases number of correct predictions"

"This is a scalable model with a stable algorithm over a large number of servers."

2.1 Addressed claims from the original paper

- Claim 1: *"Sentence embedding models like BioSent2Vec model can be used to identify similarities between patients' symptoms (and any other information like preliminary diagnosis, clinical notes, etc.) to predict diagnoses."* - The sentence embedding model doesn't successfully predict disease diagnosis.
- Claim 2: *"Increasing the size of dataset will increase the precision and recall as well as increasing value of top-K increases number of correct predictions"* - Bigger dataset (100, 250, 500, 1000, 2000, 5000) resulted in lower accuracy, precision, and recall values.

3 Methodology

There are two pieces to this section- one of them is the extracting the required data from MIMIC-III library and other portion is running it on the model. The original paper has an ad-hoc module responsible for preparing the dataset containing symptoms, prognosis and ground truth diagnosis. Due to lack of this ad-hoc module, we tried to develop a module which will filter the required columns from MIMIC-III database. This module reads data from a MIMIC-III csv file, filters required columns and

samples up to 5000 patient information. Following it, the module extracts keyword embedding using keybert, a BERT based keyword extraction technique and at the end filters for all the unwanted details such as special characters and medical measurement values and stores into a csv file. The run time for reading records from MIMIC-III database, applying BERT embedding and saving extracted symptoms to a file takes 12.5 hours for 4486 patient records on Intel i7 CPU without GPU assistance. We used Google Colab GPUs to run these steps and it took around 3.5 hours to extract 4486 patient information from 5000 samples.

The other part is the original code, which is written in Cython programming language, a super-set of Python language. We have converted Cython code into Python code which affects run-time of the code. We used a personal computer with nominal CPU (2.6GHz 6-core Intel i7) with no dedicated GPU to run the code and it should take very little time to complete the run. Please note that the algorithm has a complexity of $O(n^2)$.

3.1 Model descriptions

A Clinical Decision System aims to support the physicians in diagnosis and treatment of patients based on similarity of symptoms and diagnosis between patients. The suggested model relies on the specialized concept of a digital twin- a digital twin is a digital copy of any living or non-living object, entity, or matter. The model leverages EHR data to infer patients' diagnosis at the time of discharge. The architecture of the model consists of three layers- First responsible for preparation of medical data followed by an AI layer responsible for integrating health information and intelligent applications. The High performance Computing layer consists of a medium to handle and process the large amount of data. This layer includes CPUs and GPUs to aid faster processing. The model automates the process of gathering useful patient information at the time of admission, including medical history, symptoms, and preliminary diagnosis to identifying patient similarity and predict diagnosis of the patient at discharge. Patient similarity is established by defining similarity in symptoms and diagnosis between patients is a novel approach for patient-similarity-based methods. Integrating the pairwise patient's clinical features into an input vector and patient similarity as output vector, the challenge of multi-label classification is reduced to a single-value regression problem. The model diagnoses semantics by exploiting sentence embedding, NLP, and neural networks.

3.2 Data descriptions

The data used in this study is a digital patients library built by combining the EHRs with external knowledge obtained from social media and sensor data. The traditional data of EHRs obtained for this study is the popular MIMIC-III database developed by the MIT Lab. This de-identified database combines complete clinical data of patients admitted to the Beth Israel Deaconess Medical Center, ICU unit in Boston, Massachusetts between 2001 and 2012. It includes a vast list of features including laboratory tests, vital signs, demographics, etc,.

Each patient is represented by a feature vector represented in ad hoc data structure defined as follows: Patient, p = (id, s, pd, ssv) where,
id = patient identifiers,
s = a list of symptoms (s1, s2,...ss),
pd = a list of preliminary diagnosis (pd1, pd2,. . ., pds), and
ssv = a list of semantic symptom vectors (ssv1, ssv2,. . . ,ssvs) corresponding to the symptom list, s, and preliminary diagnosis list, pd.

A similarity measure is constructed from the patient feature vector and made consistent with the physicians' feedback in terms of whether two patients are clinically similar or not. An ad hoc extraction module has been implemented for symptom extraction since the symptoms are not explicitly specified in the database.

Preliminary diagnosis recorded in the admission notes are probable hypotheses and may not be adequate. Conversely, after discharge diagnosis noted after treatment are accurate and definite and are confirmed by the doctors.

3.3 Hyperparameters

The three major parameter affecting result are *alpha*, the symptoms similarity threshold, *beta*, the diagnoses similarity threshold and *k* the match within top k results is considered to be success. The paper discusses ideal range of values of these parameters to be used.

Other parameters like K-folds, train-test split, etc., are specified in the code itself under module Utils/Constants. Also, the model uses BioSent2Vec (based on Sent2Vec) a pretrained model which is already set with optimized set of parameters.

3.4 Implementation

The first step of creation of patient feature vectors, which is the input feature matrix, is constructed by integrating two parts - symptoms and preliminary diagnosis. The following step is semantic corpus generation through different medical knowledge. The third step is neural network building and training. Next is construction of distance-based similarity profiles. Similarities are calculated based on cosine distances between vectors. And the final step is patient prediction based on similarities from historical admissions. All this is performed for 10 folds using K-folds cross validation technique.

3.5 Computational requirements

The model that is used in this paper is based on BioSent2Vec (based on Sent2Vec) which is a pretrained model used for sentence and word embeddings and is very efficient in terms of computational complexity. It takes O(1) vector operations per word-processed both during training and inference of the sentence embedding.
80-20%(M train, N test) fold, F folds
Complexity is O(M * N * F)

The computationally heavy portion is extraction of patient symptoms, which takes over 12 hours of CPU time to extract records of 4500 patients. However, this time can be reduced by using GPU processor. We tried the same in Google Colab and it took us around 3.5 hours of GPU time to extract the same information as above.

Running model is computationally heavy and it requires a $O(n^2)$ time to finish diagnosis prediction. For example it took CPU time of 8 minutes for predicting diagnosis for 129 patients, 15 minutes for 170 patients, 25 minutes for 250 patients, over 1 hour for 500 patients and over 5 hours for 1000 patients respectively.

4 Results

The overall consensus from our experimental evaluation of the reproduction study is that the model developed by the author is not yet a good candidate for CDS framework system. There are still improvements required in terms of run-time and other performance parameter. We were successfully able to reproduce the results from the given sample data in the code, but based on our experiments the general notion of the reproduction study is in against of the hypothesis claimed paper.

4.1 Result 1

Although the BioSent2Vec model based algorithm identified few true positives, it produced greater number of false positives and thus was largely unsuccessful in identifying patient similarities from patient information like preliminary diagnosis, symptoms, clinical notes etc. Precision, recall and prediction rates are quite low as well making the system not viable. Link to Results

4.2 Result 2

Precision, recall, and accuracy worsened as the dataset grew. The model performs comparatively better within 1% difference in terms of accuracy, precision and recall values for the sample data of 129 patients provided by the author in code. However, for the extracted samples of size 100, 250, and 500 the accuracy drops considerably. Precision also decreased with increasing size of datasets. Link to Results

4.3 Additional results not present in the original paper

Due to lack of time and computing resources, we were not able to perform more ablations as we had initially decided. We were able to run the model on datasets of different sizes (100, 250, 500, and 1000) and could confirm that the experimental results of this study do not support the claims of the paper.

5 Discussion

The paper proposed a new CDS system capable of integrating heterogeneous data and drawing patient similarity from it, thus aiding physicians in diagnosis prediction. Overall, the paper is well explained, code is provided and the readily available MIMIC-III data is used in the paper. Also, the paper comes with 129 sample data. The paper is reproducible only with sample data provided. The author has written code in Cython, thus little difficulty may be faced in running the code. Due to missing ad-hoc module, extraction patient symptoms from MIMIC-III database required some trial and error to figure out exact format of dataset required to run the model. Also, converting code from Cython to Python results in code breaks at few locations which we figured out on the fly. Although, this language conversion became a hindrance since the Python code requires longer run times and code breaks could result in loss of progress.

Due to limited time and computational limitations, we were not able to perform more ablation studies or perform additional experiments with significantly larger datasets as we had decided previously. Even with this limited experimentation, we conclude that the paper is only reproducible with the sample dataset provided in the code and additional conveyed experimental results are in not favour of the claims in the paper. Running the model for different sizes of datasets shows that the model is not a good fit for CDS system.

5.1 What was easy

The easiest portions of the reproduction were obtaining the famous MIMIC-III dataset as well as obtaining the pretrained BioSent2vec model, which was the core essence of the paper.Since, BioSent2vec Model is already trained on medical corpus (PubMED MIMIC-III), it performs much better than other sent2vec models available and also reduced the time required to train the sentence embedding model. Further, the author's code was readily available in Cython language, which lowered the difficulty of reproduction study.

5.2 What was difficult

The difficult part of the study was to pre-process the MIMIC-III database in required form to be used by the code. Also, we turned the author's code from Cython language to Python language which added to the complexity of the problem statement. However, this might not be a difficulty for someone with a sound background in Cython language. Preprocessing the MIMIC-III database into desired format takes significant time than anticipated. Installing supporting libraries such as flair, keybert, and Sent2vec on a windows machine took considerable efforts since the documentation does not readily provide answers to errors thrown. Running the model on Google Colab was not helpful since Google Colab has format of Jupyter Notebooks and it reaches RAM capacity easily.

5.3 Recommendations for reproducibility

The original work (code) is done in Cython programming language, which is a super-set of Python language. We have converted Cython code into Python code which affects run-time of the code. If the reproducer is fluent in Cython, We would recommend coding in Cython itself. Also, the code is missing ad-hoc module required for pre-processing MIMIC-III data into desired format. We would suggest checking the Symptoms-Diagnosis.txt file for the format of data required running the model. Because of support of Python libraries are better in Linux, it is suggested to use a Linux based system for reproduction study and also utilize any hardware acceleration possible.

6 Communication with original authors

We did not contact the authors since the code and data were partially available and could be extrapolated.

7 References

1. C. Comito, D. Falcone and A. Forestiero, "AI-Driven Clinical Decision Support: Enhancing Disease Diagnosis Exploiting Patients Similarity," in IEEE Access, vol. 10, pp. 6878-6888, 2022, doi: 10.1109/ACCESS.2022.3142100.
2. https://github.com/ncbi-nlp/BioSentVec
3. https://cython.org/
4. https://physionet.org/content/mimiciii/1.4/
5. http://staff.icar.cnr.it/diseaseDiagnosis.zip