Final Project Draft

Rezana Ganie and David Alejandro Henriquez  
[rganie3@gatech.edu](mailto:rganie3@gatech.edu) and [dhenriquez3@gatech.edu](mailto:dhenriquez3@gatech.edu)

***Abstract—***Sample Max 5 pages excluding references and appendix

# Introduction

A clear, high-level description of what the original paper is about and what is the contribution of it.

The paper chosen to reproduce is AI-Driven Clinical Decision Support: Enhancing Disease Diagnosis Exploiting Patients Similarity. In this paper, the authors intend to create a Clinical Decision Support System (CDS) that identifies diseases and suggests treatments. Besides, the CDS uses heterogeneous data from multiple sources. For this work, the author focuses on creating a model that predicts diagnosis based on patient symptoms and preliminary diagnosis.

The model constructed consists of 5 steps. First, build a patient feature vector composed of symptoms and preliminary diagnosis from EHR data. Second, generating a semantic corpus through different medical knowledge (30 million documents from scholarly articles in PubMed and clinical notes in the MIMIC-III Clinical Database). Third, a single-layer Neural Network is used to create word embeddings. The library BioSentVec (an unsupervised version of Fast-Text, and an extension of word2vec (CBOW) to sentences) was used in the last two tasks. Fourth, once the vector representations of each patient's symptoms and diagnosis are created a similarity profile is constructed (a cosine-type similarity map is used). Finally, the fifth step is creating the predictions. In the fifth step, the top k predictions are recorded within a threshold alpha for symptoms and beta for diagnosis.

# Scope of Reproducibility

List all hypotheses from the paper you will test and corresponding experiments you will run.

The results we intend to reproduce are the results concerning the disease prediction based on patient similarity, more precisely the hypothesis that the approach is effective and accurate. The results to reproduce are:

* The precision metric results for different alpha and k.
* The recall metric results for different alpha and k.
* The number of correct predictions for different k and dataset size.

In principle, the model would be fed with the sample data, with the objective of expanding to the whole MIMIC-III dataset later.

# Methodology

## Model Descriptions

Model description

– Model architecture: layer number/size/type, activation function, etc

– Training objectives: loss function, optimizer, weight of each loss term, etc

– Others: whether the model is pretrained, Monte Carlo simulation for uncertainty analysis, etc.

The pipeline of the model consists of fourth steps: The patient's symptoms and the diagnosis list generator. The feature symptoms list vectorization through the word embedding model BioSent2Vec. The similarity profile construction with the semantic similarity cosine-type function. For a given alpha (symptoms similarity threshold), beta (diagnosis similarity threshold), and k (top k predictions) predicting the patient diagnosis. Finally, the computation of the metrics (precision, recall, and F1-measure).

It is important to recall that the word embedding task uses a pre-trained BioSent2Vec model.

## Data Descriptions

– Source of the data: where the data is collected from

provide the link if possible; if data is synthetic or

self-generated, explain how.

– Statistics: dataset size, cross validation split, label

distribution, etc

– How do you use the data: change the class labels,

split the dataset to train/valid/test, refining the

dataset

The main data source is the MIMIC-III Clinical Database. For the preliminary version of the project, sample data (provided by the article authors) from the MIMIC-III dataset is used.

The sample data consists of 1290 patient admissions with a list of symptoms and diagnoses for each. Besides, this data is spitted 90% for training and 10% for testing.

The final dataset consists of 58976 patient admissions with a list of symptoms and diagnoses for each. This dataset is split into training (80%) and testing (20%).

For each of the datasets, the data is split randomly, and the experiments are performed using 5-fold cross-validation.

## Computational Implementation

– Report the software and hardware implementation (What is your basic coding framework, PyTorch, Tensorflow, etc? What kind of CPU or GPU do you use?)

– Report hyperparameters including learning rate, dropout rate, number of iterations, training time, etc.

The project was developed in Python using CPU.

## Code

– Which parts are developed by yourself? Which

parts are referred from the codebase in original

paper or other resources?

– Provided link to your repo (Github, Gitlab,

Bitbucket, etc). Your repo should include detailed documents (README file) telling readers:

∗ Dependencies (which packages are required)

∗ Download instruction of data and pretrained

model (if applicable)

∗ Functionality of scripts: preprocessing, training,

evaluation, etc.

∗ Instruction to run the code

# Results

Report results for all experiments that you run:

– specific numbers (accuracy, AUC, RMSE, etc)

– figures (loss shrinkage, outputs from GAN,

annotation or label of sample pictures, etc) Comparison with the hypothesis and results from the original paper.

# Discussion

The discussion

Make assessment that the paper is reproducible or not. • Explain why it is not reproducible if your results are

kind negative.

• Describe “What was easy” and “What was difficult”

during the reproduction.

• Make suggestions to the author or other reproducers

on how to improve the reproducibility.

# References

1. Joyner, D. A., Ashby, W., Irish, L., Lam, Y., Langston, J., Lupiani, I., Lustig, M., Pettoruto, P., Sheahen, D., Smiley, A., Bruckman, A., & Goel, A. (2016). Graders as Meta-Reviewers: Simultaneously Scaling and Improving Expert Evaluation for Large Online Classrooms. In *Proceedings of the Third Annual ACM Conference on Learning at Scale*. Edinburgh, Scotland.

# Appendices

You may optionally move certain information to appendices at the end of your paper