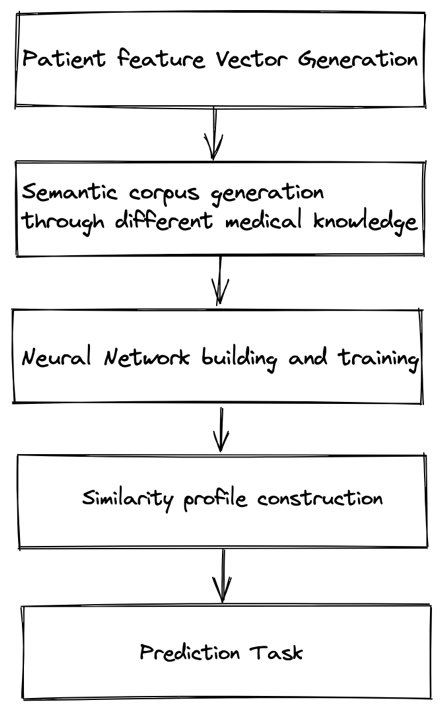
Final Project Draft

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

# Introduction

The paper we selected to reproduce is titled; 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 framework that physicians can use to that predict a new patients diagnosis based on historic patient symptoms and preliminary diagnosis. The goal of this paper is to build a fast and accurate CDS that is better than existing CDS’s which limits predictions by only considering a single patient medical condition or are built on decision rules.

The paper proposes a CDS framework which consists of 5 steps. First, build a patient feature vector composed of symptoms and preliminary diagnosis from EHR as sentences.

Second, generating a semantic corpus from MIMIC-III Clinical Database.

Third, a single-layer Neural Network is **used** to create word embeddings. This was not created as part of the framework but rather the paper uses a library: BioSentVec (an unsupervised version of Fast-Text, and an extension of word2vec (CBOW) to sentences). The reason for this includes the trade-off (speed and complexity) between simple word embeddings and RNNs, LSTMs. BioSentVec produce 700 dimension vectors for sentences (each symptom).

Figure CDS Framework

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 making a prediction by taking the input of a symptoms computing the similarity historic profiles. In the fifth step, the top k predictions are recorded within a similarity threshold alpha for symptoms and beta for diagnosis.

The motivation behind this is “In contrast to more complex neural network based models, one of the core advantages of the proposed technique is the low computational cost for both inference and training” [1]

Problem statement

“Given a patient p, a set of symptoms s felt by the patient and a set of preliminary diagnosis pd , the objective is to predict the discharge diagnosis d for p by exploiting the similarity in terms of symptoms and preliminary diagnoses with other patients already treated and for which a discharge diagnosis has been already formulated.” [1]

“The multilabel classification problem is converted to a single-value regression problem by integrating the pairwise patients’ clinical features into a vector and taking the vector as the input and the patient similarity as the output.”

“It is worth pointing out that, differently to previous approaches, the diagnosis prediction problem is addressed differently since the aim of the work proposed in this paper is to predict the exact diagnosis in terms of semantic meaning. In fact, instead of referring to the diagnosis in terms of the International Classification of Diseases (ICD-9) codes we consider the diagnosis semantics by exploiting Natural Language Processing”

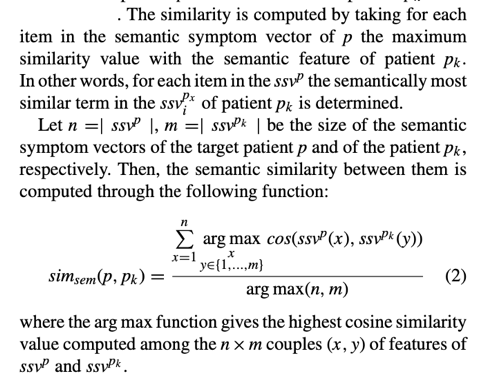
“This way we are able to implement semantic enhanced diagnosis prediction. In particular, we exploit word embedding models since neural networks shown to have the ability to learn complex semantic feature representations”

“Our objective is to design a similarity measure that operates on patient feature vectors and that it is consistent with physician

feedback in terms of whether two patients are clinically similar or not. Accordingly, the semantic similarity metrics is a key aspect in the disease prediction approach.”

# Scope of Reproducibility

The results of the experimental evaluation performed with a twofold goal:

* The first one is assessing accuracy via the CDS of the proposed diagnosis prediction approach
* The second goal included assessing how scalable the distributed discovery services can be.

The scope of reproducibility that we committed to include:

1. Creating a similar CDS framework
2. Building a **scalable** system, measured by adding more data and being able to perform the same functions.
3. Implementing a cosine similarity comparator.
4. Using the BioSentVec trained model to create symptom vectors.
5. Performing experiments on a **accuracy metric** from testing data.
6. Reuse of the data extracted from the MIMIIC dataset used by the paper for training and testing.

The results we intend to reproduce are: the hypothesis that the approach is effective and accurate.

Effectiveness:

* The accuracy metric results for different alpha and k.

Scalability:

* The scalability testing includes adding more folds of data.

The model would be fed with increasing amount of sample data expecting same level of accuracy will be met.

# Methodology

## Model Descriptions

The CDS Framework builds a pipeline and re-uses a BioSentVec Model. The model applies sent2vec to compute the 700-dimensional sentence embedding and uses the bigram model and set window size to be 20 and negative examples 10.We create feature symptoms list vectorization through the BioSentVec word embedding model.

## Data Descriptions

The main data source is the MIMIC-III Clinical Database which consists of 58976 patient admissions with a list of symptoms and diagnoses.

For the preliminary version of the project, sample data (provided by the article authors) as a subset of MIMIC-III dataset is re-used by us. We converted the data from plain text to csv formatting for use in our CDS.

The data was labelled with ICD9\_code\_diagnosis\_sentences: [list\_of\_symptoms]. The data did not use correct English Language rules. An example of a row of data: 110244\_Ocl crtd art wo infrct: Gout NOS, Old myocardial infarct.

The sample data used 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 data model: Each patient is represented as a feature vector. The vector is word embedding from symptoms, lab tests and preliminary diagnoses. These are all integrated into a single patient feature vector for every patient.

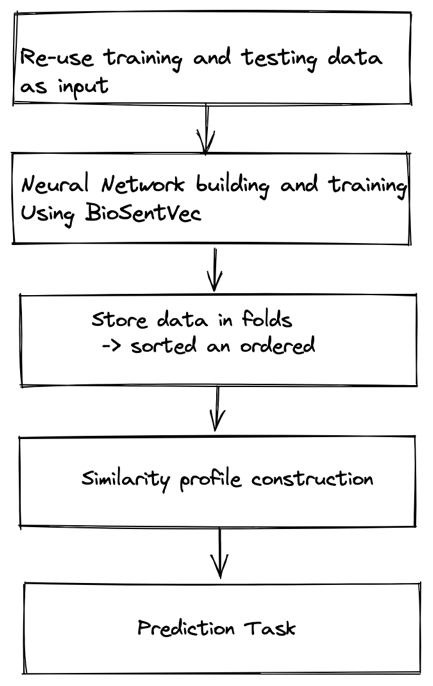
## Computational Implementation

The main goal of the project was to develop an effective and efficient CDS that produces accurate results with less complexity and time and space costs as compared to pre-existing CDS. This was achieved through intelligently designing a scalable framework that can process huge amounts of data efficiently.   
The framework included intelligently storing historic data vectors in order of related closeness and later fetching this with few queries/reads. Therefore the project was developed in Python and python libraries such as: Pandas, CSV, Itertools, Bisect, NLTK. We used CPU to prove that this can be an efficient solution.

Hyperparameters used :

* k = equaled the number of top similarity matches [1, 5, 10, 15, 20, 25, 30]
* folds = batches of training and testing data [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
* alpha = thresholds used in acceptance of similarity distances between vectors [0.7, 0.8,0.9]

## Code

Our CDS framework consist of five steps which are fairly similar to the authors In step 1 we re-used authors data but we also convert the input format into csv.  
Step 2: We used the word embedding BioSentVec model to vectorize patient symptoms and persisted these in their original folds.   
Step 3: This was a further enhancement to efficiency where we stored sorted and ordered vectors of data for quick retrieval.   
Step 4: Implement a cosine similarity calculation. This was constructed to compare test patients’ similarity against historic training patients. The selection criteria included shorter distances between the two, using various threshold criteria.   
Step 5: Finally performs the predictions and computes our accuracy metrics

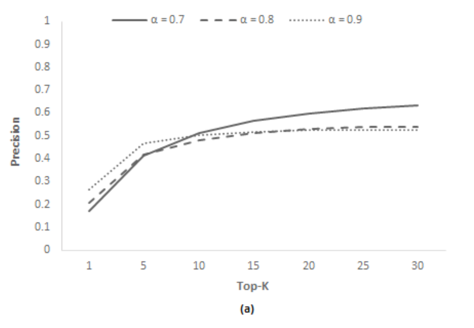
The code is written in python and all necessary dependencies are contained in a environment.yml which you can view on our repo: [Link](https://github.com/Rezana20/bd4h-group-project/tree/main/project_draft/code) to code. The readMe file contains detailed description of how to run the code.

# Results

Regarding the hypothesis for “Effectiveness:“: We attempted was to determine precision of the CDS Framework we built. We used Precision calculated as:

P= TP / (TP + FP). TP is the detection of ground truth diagnosis and FP is the number of ground truths mistakenly predicted. We calculate Precision at k. K is the number of top predictions this means the closeness between vectors.

As a result we were able to extract an average precision curve for varying values of top k most similar patients. In Figure 1, we observed precision increases as we increase the k value ( that is include more patients in our decision). We used one fold for this testing, this included: 15 test patients and 116 training patients.

Chart, line chart

Description automatically generatedThe paper produces precision using alpha [0.7, 0.8, 0.9] and performs the testing on all 10 folds of data (that is 2000 predictions). The paper performs 5-fold-cross validation. See graph on the right. The difference in results is due to us averaging of precision and using less data.

Regarding the hypothesis for “Scalability“: This will be proven through adding more data in the next delivery.

# Discussion

The paper was fairly difficult for us to reproduce for the following reasons:  
The explanation of how the framework was implemented was not clearly defined so we had to make many assumptions of the CDS.  
The similarity equation was not simple to understand as they used the words cosine similarity but implemented their own variation of what we understand by cosine similarity. (Similarity = (A.B) / (||A||.||B||) )  
They state in the paper that they create a Neural Network, which is misleading because they used a pre-trained model.  
Although they had code it was not written using good software engineering principals making it fairly hard to read and understand and having the code posed zero advantage to our implementation.

Building a scalable application was easy to understand but it is time consuming. The CDS framework proposed was logical and the steps was easy to piece together however exact reproduction was hard because of lack of documentation on the code and lack of detailed explanation of each step in the framework.

Improvements could be more documentation and enhancing the ReadMe.  
For future deliveries we will increase the size data used, we will test use varying alphas and attempt another metric to prove our two hypotheses.

# References

1. Comito, C., Falcone, D., & Forestiero, A. (2022). AI-driven clinical decision support: Enhancing disease diagnosis exploiting patients similarity. IEEE Access, 10, 6878–6888. https://doi.org/10.1109/access.2022.3142100