Paper : AI-Driven Clinical Decision Support: Enhancing Disease Diagnosis Exploiting Patients Similarity, IEEE Access 2022, C Carmela et al

Introduction

The vast amount of health data generated from sources like electronic health records (EHRs), medical images, lab results, and wearable sensors presents an opportunity to provide real-time, personalized healthcare insights. Traditional approaches in Clinical Decision Support (CDS) systems often focus on single-disease predictions for individual patients, but real-world cases often involve multiple conditions and complex interactions within patient data.

This paper proposes a new CDS framework designed to handle the challenges of integrating and analyzing diverse data sources. By using patient similarity as a core concept, the system compares patients based on the contextual relationships between symptoms and diagnoses rather than relying solely on isolated data points. This approach, enhanced by word embeddings generates rich, context-based representations of health information, enabling more accurate and comprehensive diagnosis predictions. Testing this framework on real-world medical data, such as the MIMIC III dataset, demonstrates its effectiveness, showing the potential for AI to drive impactful improvements in patient care through predictive diagnostics.

Methodology - Model Description:

The proposed CDS relies on creating "digital patient" profiles by combining traditional health data sources like EHRs with additional knowledge bases such as sensor data and social media. Relevant features like symptoms, lab tests, and diagnoses are extracted and represented in feature vectors for each patient, allowing for a rich, contextual understanding of the patient’s health. The aim is to build patient representations that capture important clinical details, while reducing the vast amount of data into a structured format usable by AI models. Using this structured feature set, the model measures patient similarity based on both symptoms and preliminary diagnoses to predict future diagnoses.

In this implementation, patient feature vectors are mapped to a "semantic space" using sentence embeddings, so that patients with similar health profiles can be identified. Unlike the original setup which used BioSentVec for these embeddings, here we use SentenceTransformers, a faster and more

lightweight alternative. Note all the models finally form embedding vectors for each patient, and similarity scores between these vectors is formed using the commonly used cosine formulae.

Data Description:

This model is tested on the MIMIC-III database, which covers data from thousands of patients and includes both admission and discharge diagnoses. For the similarity-based prediction approach, a core challenge is accurately modeling these diagnoses and symptoms to reflect their true meanings without relying on ICD-9 categorical codes. The data for this is obtained specifically from *noteevents* table in *mimiciii_notes*. Note k-fold cross validation is used as the CV strategy on the training data (initial data split on a 80-20 proportion) where k=5. The dataset size is about 129 patients' symptoms & diagnoses.

In addition, a semantic corpus is built to further enhance diagnosis prediction. Originally, BioSentVec, trained on millions of medical texts was used to generate word embeddings. In this version, we use SentenceTransformers for reduced resource requirements.

Scope of Reproducibility

In reviewing the original paper's reproducibility, it's evident that the authors have left a substantial gap in terms of structured, accessible code and documentation. There was no GitHub repository, instead we were left with a highly fragmented codebase—approximately 20,000 lines of Cython code with a handful of demo data scattered across files. The authors made use of legacy Python libraries that lack compatibility with current packages, compounding the problem by not including a requirements.txt file. This omission led to numerous dependency conflicts, easily avoided with basic containers.

Initially, I tried using a similar setup with sent2vec, but lack of guidance and substantial training times proved unsustainable. I considered mimicking their Cython implementation to improve runtime, but without a clear structure, replicating the exact functionality proved infeasible. Finally, I replaced sent2vec with the SentenceTransformers model, still capable of generating rich embeddings from large text corpora. From SentenceTransformers, I tried several pre-trained models -

1. **all-MiniLM-L6-v2** – A nice practical model but this model is not tailored specifically to medical text, and we are not leveraging a GPU setup or custom Cython code

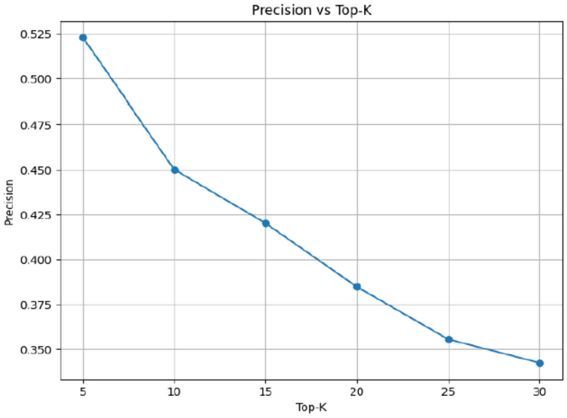
2. **all-mpnet-base-v2** – This was the strongest model I tested – took a few hours to encode and run, but still not producing as good results

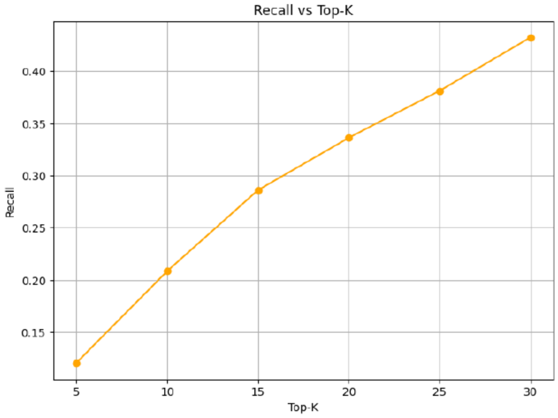
3. **Bio_ClinicalBERT** – This is a very strong model pre-trained on a huge corpus, importantly fine-tuned to medical data

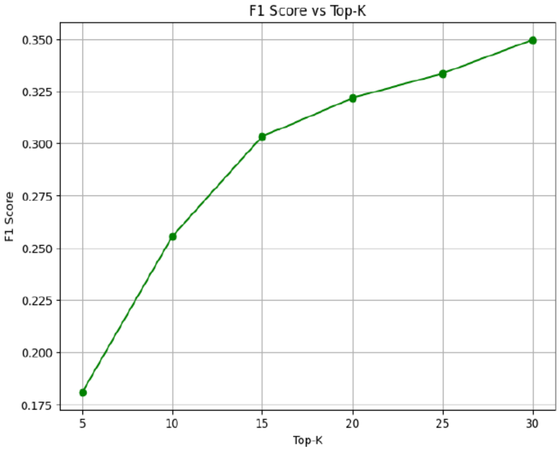
Results

I will show results here for two of my best NLP models' runs –

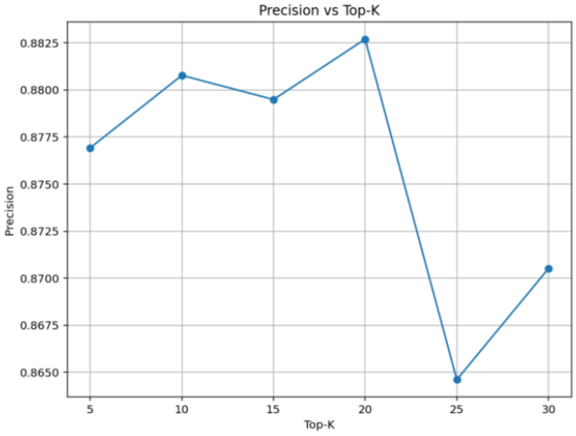
All-MiniLM-L6-v2 -;

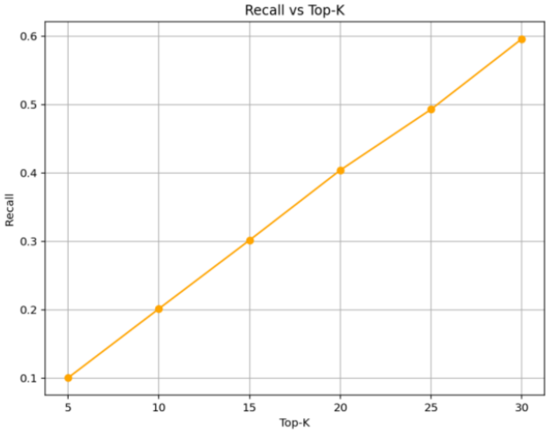


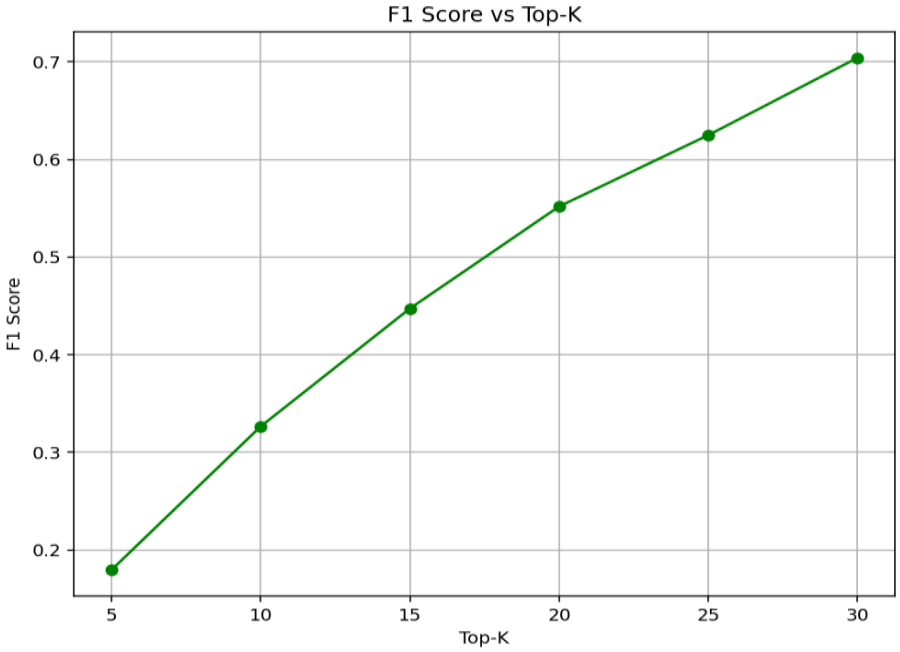




Bio_ClinicalBERT (Fine-tuned for MIMIC 3 Data)







The results we achieved in the first, as shown in the plots, are already surprisingly quite decent – however it is the results from the 2nd model aforementioned that produces the really staggering results, on par with the paper's results if not slightly bettering it. This outcome is somewhat unexpected, especially given our simplified approach and the fact that we opted to use SentenceTransformers rather than the original sent2vec model on a domain-specific corpus. Our use of a generic, pre-trained sentence embedding model and biomedical corpus trained **clinicalBERT** still yielded competitive precision, recall, and F1 scores, indicating that the methodology of predicting diagnoses based on symptom similarity retains value even without the high computational demands of the original setup. Additionally fine-tuning **clinicalBERT** model on MIMIC data really boosted performance.

Code & Methodology Detailed Implementation

1. Collect the symptoms data of patients from MIMIC 3 – split into train, test datasets

2. Load the NLP embedding model – fine tune for general MIMIC Data

3. Embed the symptoms in training set for patients' symptom similarities

4. Use k-fold cross validation on train data to determine **optimal similarity threshold**

5. Looping through each test data point, find subset of similar patients by filtering for similarity scores higher than the threshold above, ranking them in order of scores

6. For each of k=[5,10,15,20,25,30] - find the top k such similar patients and predict the final diagnosis to be the same as those for these k patients

7. Compute precision, recall & F1 scores based on diagnoses similarity scores of these k predictions for whole test set and plot the graphs

Note final similarity threshold chosen is 0.9 - the code is run on a colab notebook and can be found in *https://github.com/ksingla-GL/NLP_Disease_Diagnosis_CDS_Prediction/blob/main/BD4H_Final.ipynb*

Discussion – As a final note, I would say the basic modelling techniques used here are well explained and were thus reproducible, however the incomplete unstructured data processes with huge amounts of compute time & power requirements were largely unexplained and made technical reproduction using the same NLP model infeasible. Fortunately using an alternative lightweight fine-tuned NLP model almost yielded the same performance – which makes one wonder if the authors could have better spent their time in refining the methodology rather than writing tens of thousands of lines of code in Cython.