AI FOR HEALTHCARE: DIAGNOSIS, CLINICAL-TRIAL MATCHING, AND PATIENT RECRUITMENT

A Thesis

by

KARIMI ABHISHEK DAS

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Anxiao Jiang
Committee Members,	Ruihong Huang
	Xiaoning Qian
Head of Department,	Scott Schaefer

August 2020

Major Subject: Computer Science

ABSTRACT

Medical diagnosis is the most critical component in the treatment of a patient. But diagnosis often is a complicated process since a myriad of diseases share the same symptoms. If a patient is diagnosed with a disease in its end-stage, potential new treatments (clinical trials) are sometimes the last option available. However, matching a patient to the correct clinical-trial requires advanced medical knowledge on behalf of the patient.

In this study, we try to address the following problems and close the technical gaps, (i) *Diagnosis*: Advances in neural network approaches and the availability of massive labeled datasets have sparked renewed interests in automated diagnosis. We explore novel techniques to identify pathology in chest radiographs by using a labeled radiograph dataset, which is also substantially large for the domain of medical diagnosis. (ii) *Clinical-Trial Matching*: Given the difficulty of perusing the jargon in standard clinical trial texts, we try to complement the process by using machine learning and information retrieval methods to fetch similar health records showing the entities responsible for the match.

We implement an efficient visual tool (*TextMed*) to aid our algorithm and make it easier for users to utilize the power of machine learning. Our tool helps in searching through a database of criteria and records and fetches the information about the query.

DEDICATION

To my mother, father, sister, and friends. Without their support through thick and thin, this wouldn't have been possible. Thank you to my academic advisor Prof. Jiang who guided me in this process and the committee who kept me on track.

ACKNOWLEDGMENTS

I would like to express my deepest appreciation to my advisor Prof. Jiang, who played a pivotal role by giving constant feedback and guiding me throughout. I would also like to extend my thanks to Pulakesh Upadhyaya for being a great mentor and giving me insightful inputs on my research and also helping me in day to day graduate life.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a thesis (or) dissertation committee consisting of Professor Anxiao Jiang and Professor Ruihong Huang of the Department of Computer Science and Engineering and Professor Xiaoning Qian of the Department of Electrical and Computer Engineering.

The data analyzed for Diagnosis studies (CheXpert) was published by Stanford University. The inclusion and exclusion criteria dataset have been made publicly available by the NIH. Since real Electronic Health Records are subject to high privacy scrutiny and challenging to obtain even when anonymized; we used fake Electronic Health Records generated by Synthea tool. All other work conducted for the thesis (or) dissertation was completed by the student independently.

Funding Sources

Graduate study was supported by Teaching Assistantship from the Department of Computer Science and Engineering at Texas A&M University.

NOMENCLATURE

AI	Artificial Intelligence
BERT	Bidirectional Encoder Representations from Transformers
BM25	Best Matching 25
BioBERT	Bio Bidirectional Encoder Representations from Transformers
CNN	Convolutional Neural Networks
CheXpert	Chest X-Ray Dataset
Double-DIP	Double Deep Image Priors
EHR	Electronic Health Record
GLUE	General Language Understanding Evaluation
GRU	Gated Recurrent Unit
IR	Information Retrieval
LSTM	Long Short-Term Memory
MIMIC-III	Medical Information Mart for Intensive Care III
MLM	Masked Language Model
MeSH	Medical Subject Headings
MedNLI	Medical - Natural Language Inference
NDCG	Normalized Discounted Cumulative Gain
NIH	National Institute of Health
NLI	Natural Language Inference
NLP	Natural Language Processing
RL	Reinforcement Learning

SBERT	Sentence Bidirectional Encoder Representations from Transformers
SQL	Structured Query Language
SciBERT	Scientific Bidirectional Encoder Representations from Transformers

TABLE OF CONTENTS

Page

ABSTRACT ii

DEDICATION iii

ACKNOWLEDGMENTS iv

CONTRIBUTORS AND FUNDING SOURCES v

NOMENCLATURE vi

TABLE OF CONTENTS viii

LIST OF FIGURES x

LIST OF TABLES xi

1. INTRODUCTION 1

    1.1 Medical Diagnosis 1
    1.2 Clinical-Trial Matching 1
    1.3 Artificial Intelligence 2
    1.4 Our Work 4

2. RELATED WORK 6

    2.1 Medical Diagnosis 6
    2.2 Clinical-Trial Matching 8

3. METHODOLOGY 10

    3.1 Medical Diagnosis 10
        3.1.1 Augmentation 10
            3.1.1.1 Method 10
            3.1.1.2 Results 11
        3.1.2 Attention-Densenet 13
            3.1.2.1 Self-Attention 13
            3.1.2.2 Self Attention for Images 13
            3.1.2.3 Results 14
        3.1.3 Image Segmentation 15
            3.1.3.1 Method 15

3.1.3.2 Training and Results 17
3.2 Clinical-Trial Matching 19
3.2.1 Variable Detection 20
3.2.2 NDCG 24
3.3 Embeddings 26
3.3.1 BioSent2Vec 26
3.3.2 BERT Embeddings 26
3.3.2.1 Transfer Learning 27
3.3.2.2 BERT Structure 29
3.3.2.3 BioBERT 29
3.3.2.4 Sentence BERT 30
3.3.3 Methodology 31
3.3.4 Named Entity Recognition 33
3.3.5 BM25 + SBioBERT 35
3.3.5.1 BM25 35
3.3.5.2 Method 36
3.3.6 TextMed - A visual tool for analyzing medical text 37
3.3.7 Fake Electronic Health Records 40
3.3.7.1 Method 40

4. SUMMARY AND CONCLUSIONS 42

4.1 Medical Diagnosis 42
4.2 Clinical-Trial Matching 43
4.3 Future Work 44

REFERENCES 46

APPENDIX A. CODE 50

LIST OF FIGURES

FIGURE Page

1.1 A sample chest radiograph from CheXpert dataset 4

1.2 Frontal and Lateral radiograph of the same patient showing Pleural Effusion..... 5

3.1 ROC Curves for Augmentation 12

3.2 Attention Module used in Attention-Densenet architecture 14

3.3 ROC Curves for Attention-Densenet..... 16

3.4 Left: A sample radiograph of patient suffering from Atelectasis, Right: Image segmentation of the radiograph showing different lung sizes..... 17

3.5 ROC Curve for Atelectasis after Image Segmentation 18

3.6 Sentence BERT Architecture..... 31

3.7 TextMed tool - Interface 38

3.8 TextMed tool - Entity Recognition 39

3.9 TextMed tool - Table 39

3.10 TextMed tool - Search 40

LIST OF TABLES

TABLE Page

3.1 Distribution of classes in CheXpert dataset 10

3.2 ROC-AUC scores for Augmented vs Non-Augmented data 12

3.3 ROC-AUC scores for Attention-Densenet 15

3.4 Similar criteria to "Body mass index greater than 22.5" 21

3.5 Similar criteria to "Has cancer or a malignant tumor, treated thyroid disorder or has a history of seizure disorder" 22

3.6 Similar criteria to "Cerebral Spinal Fluid (CSF) Amyloid Beta 1-42 (A42) less than or equal to 600 pg/mL, or A ratio of total tau to A42 greater than or equal to 0.39." . 22

3.7 Similar criteria to example 4 23

3.8 Examples of variable detection 24

3.9 Similarity found using BioSent2Vec for example 4 26

3.10 Similarity found using BioSent2Vec for example 2 27

3.11 Sentence Similarity Metrics on the development dataset after training on MedNLI .. 32

3.12 Sentence Similarity Metrics on the development dataset after training on MedNLI and continuing training on our dataset 33

3.13 Model performance on Medical NER datasets 34

3.14 Distribution of n-grams in PubMed Phrases 35

3.15 NDCG Scores for various models 37

1. INTRODUCTION

1.1 Medical Diagnosis

Medical diagnosis is the process of determining which disease or condition explains a persons symptoms and signs. The information required for diagnosis is typically collected from a history and physical examination of the person seeking medical care. Diagnosis is the most critical component in the process of curing a patient. Diagnosis is often tricky because various diseases and conditions show the same primary symptoms. For example, fever and headache are prevalent symptoms accompanied by any disease; that is why it is difficult for healthcare professionals to figure out the underlying cause. Statistically speaking, diagnosis can be seen as a classification problem where each piece of information helps us to reach the actual cause and identify the correct disease or condition.

There are various tools at disposal for healthcare professionals to carry out the diagnosis procedure. One of the most commonly used tools is radiology. Radiology is the medical discipline that uses medical imaging to diagnose conditions. Various techniques are used to create visual representations of the interior of a body for clinical analysis. Medical imaging is a noninvasive procedure because no instrument is introduced into a patients body. In a restricted sense, medical imaging can be seen as a mathematical problem in which we are trying to detect the cause by the effect (the observed signal). A mathematical problem that can be identified as a classification problem can be solved using modern Artificial Intelligence methods. A sample chest radiograph is shown in figure 1.1

1.2 Clinical-Trial Matching

One of the most critical conditions today is cancer, which has mixed results for survival even after it is diagnosed. It is essential to be able to enroll patients for new clinical trials, which can cure patients if done timely and also provide insights on whether a clinical trial is useful or not. Depending on the particular conditions of a patient, the clinical trial being matched varies. Patient

enrollment for these trials remains a significant challenge. Matching and enrollment are drivers by the healthcare providers or by their proxies (e.g., relatives or caretakers). At any point in time, there are numerous trials, which are underway in the United States. A specific patient would qualify only for a small set. The trial designers design the trials in a way that they are particular, and thus the patients matched would get the most benefit from the trial. At the same time, numerous trials fail because of a lack of sufficient candidates.

In many cases, information from the patient is incomplete in the electronic health records (EHR). Patients can visit multiple healthcare systems, while the trial recruiter only has access to one of them. There is also genetic information that patients may process but not available to the recruiters. From a patient perspective, it is tough for them to understand all the technical jargon in clinical trial criteria. From a recruiters perspective, lacking the population prevalence of medical conditions makes it hard to set the scope of inclusion/exclusion criteria that would be most appropriate for their studies.

Diagnosis can be seen mathematically as a classification problem where healthcare professionals are trying to classify the observations from a particular known disease. On the other hand, clinical trial matching can be seen as a query matching problem where the patients history or record can be seen as an input query to be matched against a particular trial. Automated diagnosis and clinical trial matching can save millions of dollars of money by reducing human intervention, speeding up the process, and saving time, which can result in faster diagnosis and matching and thus saving numerous lives. Artificial Intelligence (AI) provides us a myriad of techniques to automate tasks like matching and classification. Our work explores the use of AI for diagnosing chest radiographs and clinical trial matching.

1.3 Artificial Intelligence

Computer Science defines Artificial Intelligence research as the study of intelligent agents that perceive the data from its environment and take actions that maximize its chances of achieving its goals. A more elaborate definition would be a system that can correctly interpret external data, learn from the data, and use those learnings to achieve specific goals [1]. In the current era,

boundaries in Artificial Intelligence are mainly being pushed by the use of Deep Learning. Deep Learning in its current form is a subset of machine learning algorithms that consist of multiple layers of artificial neurons and can extract high-level features from a particular dataset [2]. Modern-day deep neural networks are based on artificial neural networks where a single neuron can be viewed as a non-linear unit that takes a weighted linear combination of inputs and outputs a value after applying the non-linear activation function on the output.

Applications of Artificial Intelligence (AI) in healthcare have been explored, and more and more companies are coming up in this area. The rise of AI in healthcare is happening at a gradual and steady pace, although there are not many end-to-end systems, nonetheless that has not stopped the AI community in assisting doctors and automating various tasks involved in the pipeline starting from diagnosing to even providing treatments. Much of the current progress in AI applications to healthcare can be attributed to the fact that bigger datasets are being made publicly available by healthcare institutions and the computing power available at the hands of humanity increasing folds each year.

Current research in the application of AI in healthcare can be broadly classified into the following areas, namely Radiology, Imaging, Disease Diagnosis, Telehealth, Electronic Health Records, drug interaction, and the creation of new drugs. Deep learning has been extensively successful in the domain of images more than anything else. This success can be attributed to Convolutional Neural Networks (CNN), which was the brainchild of Yann LeCun [3]. Because of the ImageNet competition [4] there has been an explosion of various kinds of CNN architectures. These architectures improved on the accuracy, efficiency, or size of the network over the previous architectures. The progress of these architectures created a big zoo of CNN models, which could now be used for a variety of applications. Since CNNs are successful, they have been accepted widely by the industry. Most forms of image and video processing include some form of CNNs underneath. CNNs have made their way into healthcare mainly in the domain of imaging and radiology. Radiology mainly consists of identifying some form of defect or disease in particular body parts from the radiograph image. With enough supervised data, CNN can handle the task. With the advent of

radiology datasets, much CNN research has spawned in this area.

Figure 1.1: A sample chest radiograph from CheXpert dataset

1.4 Our Work

In our work, we explore a new CNN architecture for identifying the presence of pathology with the help of lung radiograph images. The dataset which has been published by Stanford University is publicly available [5]. Some sample images from the dataset are shown in figure 1.2. For one pathology, the task can be simplified even further because the discriminating criteria is the size of the lung. For the clinical trial matching task, we first devise classic algorithmic techniques that serve as our baseline. Then we explore embeddings for the medical text to find similarities between trials and patient history.

We build a tool for patients to come and receive questionnaires to help them identify particular trials. The tool helps experts and doctors to be able to save the inclusion and exclusion criteria along with the details about the criteria. The experts can choose to see the criteria they have saved, and the patients can attempt a questionnaire many times, depending upon their needs. We bring the work done in the dissertation to life by building a visual tool named TextMed. TextMed helps in

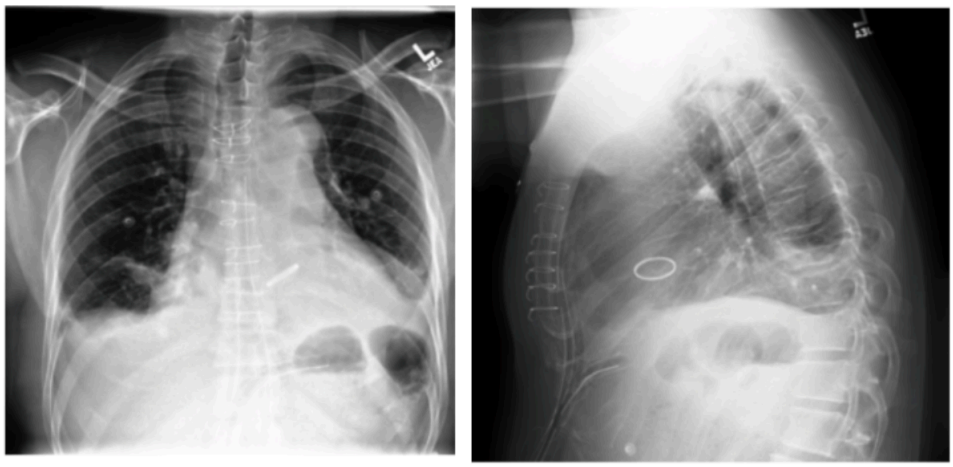


Figure 1.2: Frontal and Lateral radiograph of the same patient showing Pleural Effusion

analyzing a medical text, preferably a criterion by identifying the variables using entity recognition and further using that and machine learning model generated embeddings to find similar criteria and electronic health records. A table displays the fetched records. We give the user the ability to search and download the results as a comma-separated file.

2. RELATED WORK

2.1 Medical Diagnosis

The usage of Artificial Intelligence in healthcare has traditionally met the bottleneck of the requirement of large amounts of data. Most modern-day machine learning algorithms, especially deep learning algorithms, need a humongous amount of data to train a robust, generalizable model. Collecting large amounts of data in the medical domain is not as easy as compared to collecting images of naturally occurring everyday objects or animals. There are various regulations that the data collectors have to abide because the data is highly sensitive data related to patients, making the general data collection process painstakingly slow in some cases as well. For progress in deep learning and automated algorithms, there is a general need for datasets that conform to the following three criteria (1) the data should be extensive (2) the data has strong reference standards (3) the data provides expert human metrics for comparison. Rajpurkar et al. [5] published a large dataset of chest radiographs for automated radiograph interpretation. The dataset comprises of 224,316 radiographs of 65,420 patients. The authors have labeled each radiograph for the presence of 14 observations. Each observation can either be positive, negative, or uncertain.

The radiographs have been collected from Stanford Hospital. These radiographs have been taken between October 2002 and July 2017 in the patient centers. It was decided to consider 14 observations, which were the most prevalent in the radiographs and based on clinical relevance. The relevance conformed to the Fleischner Societys recommended glossary [6]. The No Finding observation was also included to take into consideration the case in which all pathologies are absent. To label the dataset, they used a custom rule-based labeler that operated on each radiology report to extract the pathology. It worked in three stages: mention extraction, mention classification, and mention aggregation. In the mention extraction phase, an extensive list of phrases (curated by board-certified radiologists) was used to match and extract a mention. In the mention classification stage, they used a pipeline to see if it is a negation or uncertain or positive depending

on how the words are phrased around the mention. In the aggregation phase, a final label was decided. Observations with at least one mention that is positively classified were assigned a positive, uncertain if not positively classified or has one uncertain mention and negative if it has a negative mention.

Rajpurkar et al. implemented baseline models that take the input image as a single-view chest radiograph and output the probability of each of the 14 observations. If there is more than one-view available, their model outputs the maximum probability of the observations across the views. Apart from the positive (1) and negative (0) labels, it was essential to deal with the uncertain (u) label. They explore various uncertainty approaches, which we will brief. (i) *Ignoring*: This is the most basic approach in which the uncertain (u) label is ignored, and the associated data points are not taken into account for training. Ignoring data points from a specific class can produce a biased model since the data distribution is skewed and not uniform for most of the pathologies. (ii) *Binary Mapping*: In binary mapping, the uncertain (u) label is either mapped to negative (0) or mapped to positive (1). This approach can distort decision mapping because assuming the uncertain labels to be either positive or negative holds back the classifier from learning semantic information about the data. (iii) *Self-Training*: This is a semi-supervised learning approach in which the model is trained according to the ignore approach initially and then use the model to make predictions and re-label each data point with the probability prediction outputted by the model. 3-Class Classification: As the name suggests, this treats uncertain (u) as its label, and the model is trained on that. The probability of the prediction among the classes should sum up to 1. The 3 classes are $\{p_0, p_1, p_u\}$ and $p_0 + p_1 + p_u = 1$. The loss is set-up as the mean of the multi-class cross-entropy over the observations. Multi-class cross-entropy is the most standard approach in most modern-day deep learning algorithms that classify into particular classes. We will also use the 3 class classification approach.

$$L(X, y) = \sum_{o \in \{u,1,0\}} y_o \log p\left(Y_o = o | X\right)$$

2.2 Clinical-Trial Matching

We are trying to explore how clinical trials for various patients can be matched with the right treatments. There has been active research in the domain of fetching clinical trials from queries. Most of the classical approaches use either a strict Rule Matching-based approach or Ontology-based approach or a combination of both [7]. In the strict Rule Matching-based approach, large volumes of medical articles (and criteria) are fed into ElasticSearch. The patients profile is then formulated as a query against this system, which then matches fields such as title, gender, age, criteria. The concepts for diseases, genes, age, gender, and treatments are set as must match for the queries. Other concepts are set as should match for queries. Since ElasticSearch is based on BM25 [8] algorithm, which is a classical query algorithm that returns a number for query matching, the entire query matching module returns a ranked list of criteria which are matched. The approach is the Ontology-based approach, which first filters out all the irrelevant articles/criteria. The system then enhances the information provided with topics provided by synonyms in genes, diseases, or other conditions. It further splits each disease and other concepts and links them to the nearest concept in the MeSH [9] hierarchy. After that, the process is reduced to a standard information retrieval (IR) problem where the matching score is computed using BM25, taking into account the text, gene-level data, and MeSH. Then there is the hybrid architecture, which fuses both the above approaches.

There has been active research over the years in projecting words and sentences into a vector in a d-dimensional space using which measuring similarity of new constructs (words/phrases) becomes relatively simple. One of the cornerstones works in the domain of NLP is word2vec [10], which projects a word to a dense vector in a d-dimensional space. Word2vec sparked the start of an era in NLP where the approach revolved around converting words, then phrases and sentences to vectors directly, instead of working on the tokens. New approaches in projecting a sentence to a vector involved taking the mean of the word embedding of the tokens. While this idea is simple and might work for small sentences, it will fail if there are opposite words in the sentences which cancel each other out. Moreover, taking the mean doesn't take into account the order of the tokens

in the sentence.

Another major challenge in the domain of NLP is variable-length inputs. To overcome this, researchers started using Recurrent Neural Networks (RNN). Vanilla RNN had the problem of having a short memory. Later GRU [11] and LSTM [12] (which was invented in the late 90s) were used to tackle short term memory issues. Equipped with this, researchers came up with models that converted the semantic meaning of the entire sentence to a vector. Prominent examples are sent2vec [13], InferSent [14], Universal Sentence Encoder [15]. We use a variant of Sent2Vec called BioSent2Vec [16] (specifically for medical domain) in our research. Recurrence models were slow because there was an inherent component of time, and derivatives were moved backward in time direction and prediction direction. Specialized machine learning hardware could not optimize the matrix calculations on RNNs as much as they could do with feed-forward neural networks.

Recent advances in Transformers [17], which allowed for NLP models to use feed-forward neural networks, significantly increased speed. Bidirectional Encoder Representations from Transformers (BERT) [18] which was developed by stacking these transformers and trained on two new tasks MLM (Masked Language Model): removing 15% of the words and train the model to predict the missing words based on bidirectional context and NLI (Natural Language Inference): make the model predict whether sentence B comes after Sentence A or not, provided state-of-the-art results on the GLUE [19] benchmark. BERT weights are easily transferable, and it can be easily extended to solve new tasks down the pipeline. We use BERT extensively in our research. We try to combine both the classical and machine learning approaches to find similarities between clinical trial criteria, find variables, and extract conditions in the inclusion and exclusion criteria.

3. METHODOLOGY

3.1 Medical Diagnosis

3.1.1 Augmentation

3.1.1.1 Method

The radiograph dataset published by Stanford is highly unbalanced. Unbalanced in this context means that the distribution of the data among the labels, namely negative (0), positive (1), and uncertain (u) is not uniform. Balanced data generally results in better model training. Table 3.1 shows the distribution of the dataset among the various labels.

Pathology	Positive (%)	Uncertain (%)	Negative (%)
No Finding	16627 (8.86)	0 (0.0)	171014 (91.14)
Enlarged Cardiom.	9020 (4.81)	10148 (5.41)	168473 (89.78)
Cardiomegaly	23002 (12.26)	6597 (3.52)	158042 (84.23)
Lung Lesion	6856 (3.65)	1071 (0.57)	179714 (95.78)
Lung Opacity	92669 (49.39)	4341 (2.31)	90631 (48.3)
Edema	48905 (26.06)	11571 (6.17)	127165 (67.77)
Consolidation	12730 (6.78)	23976 (12.78)	150935 (80.44)
Pneumonia	4576 (2.44)	15658 (8.34)	167407 (89.22)
Atelectasis	29333 (15.63)	29377 (15.66)	128931 (68.71)
Pneumothorax	17313 (9.23)	2663 (1.42)	167665 (89.35)
Pleural Effusion	75696 (40.34)	9419 (5.02)	102526 (54.64)
Pleural Other	2441 (1.3)	1771 (0.94)	183429 (97.76)
Fracture	7270 (3.87)	484 (0.26)	179887 (95.87)
Support Devices	105831 (56.4)	898 (0.48)	80912 (43.12)

Table 3.1: Distribution of classes in CheXpert dataset [1]

[1]Reprinted with permission from "Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison" by J. Irvin, P. Rajpurkar, M. Ko, Y. Yu, S. Ciurea-Ilcus, C. Chute, H. Marklund, B. Haghgoo,R. L. Ball, K. S. Shpanskaya, J. Seekins, D. A. Mong, S. S. Halabi, J. K. Sandberg, R. Jones,D. B. Larson, C. P. Langlotz, B. N. Patel, M. P. Lungren, and A. Y. Ng, 2019 AAAI Press, pp. 590-597, Copyright 2019 by Association for the Advancement of Artificial Intelligence.

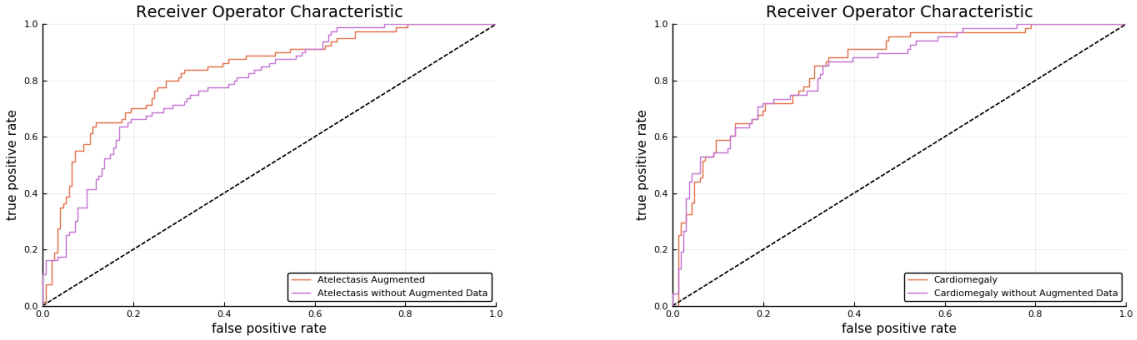
We use the standard augmentation techniques used for improving the performance of Convolutional Neural Networks by augmenting the data for the class, which has less data by providing tilts at certain angles, flipping, and certain other transformations. Since radiographs are translation and rotation invariant, this increases the performance of the model and compares it to standard methods with no augmentation. We use the python library implementation called Augmentor [20], which helps in making a pipeline that serves the rotational and translational or a combination of both of the image to the model to help the model to generalize better and find a better discriminant boundary. In our approach, we oversample the underrepresented data and thus create new data for training. Experiments were carried out using various CNN architectures like ResNet152, DenseNet121, Densenet169, and Inception-v4. It was found that DenseNet121 and DenseNet169 performed better than other architectures. Images were fed into the network after resizing as 320×320 pixels. To oversample, we zoomed the images to a scale ranging from 1.05 to 1.2, with a probability of 0.3. We subsequently rotated the image 25^o with a probability of 0.7 and then flipped alongside the horizontal axis with a probability of 0.7. We used the Adam optimizer with the default β parameters, $\beta_1 = 0.9$, $\beta_1 = 0.009$ and a learning rate of 1×10^{-4} . At each epoch, the dataset was shuffled, and batches were sampled with a size of 16. The training was done for 10 epochs. Callbacks for stopping early, when there is no improvement were given to the optimizer. The training was done on Tesla V100 GPU, and each training took roughly 7 hours to complete. Apart from the ROC-AUC scores, we also present the ROC-AUC curve. Receiver operating characteristic curve, or ROC curve is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied.

3.1.1.2 Results

The validation set contains a total of 200 studies from 200 patients. We randomly sampled these images from the dataset. Three board-certified radiologists individually annotated each image in the validation set, marking whether pathology is present or absent or unlikely. These annotations were then binarized so that the uncertain and positive cases are treated as positive and the negative cases as negative. The majority of these binarized annotations are then used as the final ground

truth.

We take two of the most important pathology and plot the receiving operator characteristic area under the curve for both of them under different methodologies. For the cases of atelectasis, the area under the receiving operator characteristic curve increases significantly from 0.787 to 0.828. The ROC-AUC curves are shown in figure 3.1.



(a) Atelectasis (b) Cardiomegaly

Figure 3.1: ROC Curves for Augmentation

Now, for the case of cardiomegaly, we have similar improvements in the receiving operator characteristic area under the curve. The area under the curve for non-augmented data is 0.837, whereas, for the augmented data, it is 0.844. We see that balancing the dataset increases the ROC-AUC score as compared to the unbalanced dataset. The above curve depicts the exact ROC curve for cardiomegaly.

Table 3.2 compiles the ROC-AUC scores for the experiments carried out.

	Cardiomegaly	Atelectasis
Non-Aug	0.837	0.787
Augmented	0.844	0.828

Table 3.2: ROC-AUC scores for Augmented vs Non-Augmented data

3.1.2 Attention-Densenet

3.1.2.1 Self-Attention

Self-Attention was introduced by Vaswani et al. [17]. Let us take a simple example sentence, *"The animal did not cross the street because it was too tired."*. What does the "it" associate to, in this example, one might ask? We can easily associate "it" to the animal. However, its tricky for the algorithm to do so. The goal of the self-attention mechanism is to predict these associations, predict that the word "it" and the word "animal" have a strong correlation and refer to the same thing here. This understanding is then used by the model to process the text and perform better.

3.1.2.2 Self Attention for Images

Now, one might naturally ask how does self-attention mechanism in the text help us with finding pathology in chest radiographs. In the case of medical images like chest radiographs for some diseases, there are various artifacts in the radiograph, which can contribute to the identification of the disease. Having this hypothesis in mind, we propose introducing an attention block inside the [21] densenet architecture, which helps in identifying correlations between different portions of the image. The core idea of the attention module used in our architecture is inspired from here [22]. The input to the densenet module is updated by applying the attention blocks (figure 3.2), which take $h \times w \times c$ as input and output $h \times w \times c_o$. The new number of output channels encode the information, which is then further passed down to the denseblocks.

In order to understand the working of the attention block applied to a 2D image, we need to follow the tensor operations. Let us assume the image to have height h and width w and the number of input channels as c. So the volume of the input is $h \times w \times c$. Our first operation is to take 1×1 convolution filters and convolve them to get three different things queries, keys, and values. Let the number of channels in the queries be c_q, keys be c_k and values be c_v. Since it is a 1×1 convolution, we have not increased or decreased the height or width of the block. Now we do the unfold operation; in the unfold operation, the input is a volume of $h \times w \times c_k$, but the output is a 2D tensor with one dimension as $h \times w$ and the other dimension as c_k. We do this operation on

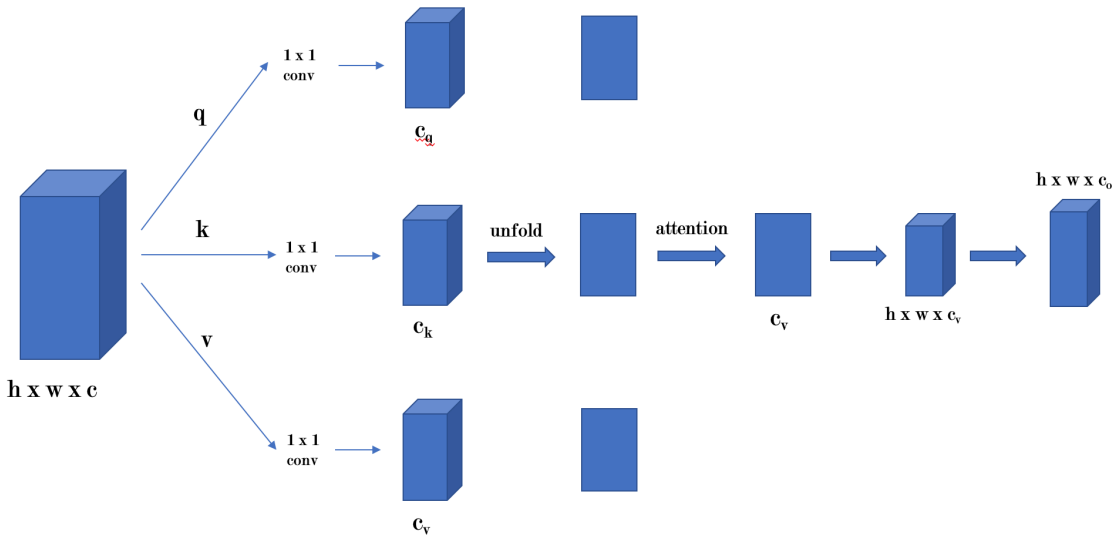


Figure 3.2: Attention Module used in Attention-Densenet architecture

each of the queries, keys, and values. We now have queries, keys, and values in a state from where we can apply the attention operation.

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

After applying the attention operation, we get the tensor whose dimensions are $h \times w$ and c_v. We do a packing operation on the 2D tensor to convert it into a volume of size $h \times w \times c_k$ and then finally apply the last linear operation to go from c_k channels to c_o channels. The packing and the linear operation, along with the standard denseblock output, is fed into the next denseblock. We use this module in tandem with the densenet121 architecture. It increases the number of parameters in the model by roughly 10%.

3.1.2.3 Results

We trained the Attention-Densenet model on one of the most critical pathologies and compiled the ROC-AUC scores in table 3.3. We can see the score has improved for some of them, but not

for all. ROC-AUC curves for a selected few pathologies are shown in figure 3.3. We hypothesize that for some diseases, the radiograph does not provide many artifacts to attend on, and hence it is not helping the classification. For these diseases, more number of parameters is harming the overall classification because of overfitting. However, in the case of some diseases, it increases the ROC-AUC score. We train these models for 20 epochs and report the best performing model over the validation set. The code has been made public on github[2].

Pathology	AUC Score
Cardiomegaly	0.838
Atelectasis	0.789
Edema	0.868
Pleural Effusion	0.768
Pneumonia	0.900
Consolidation	0.887

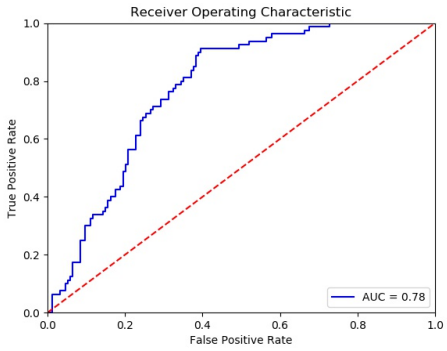
Table 3.3: ROC-AUC scores for Attention-Densenet

3.1.3 Image Segmentation

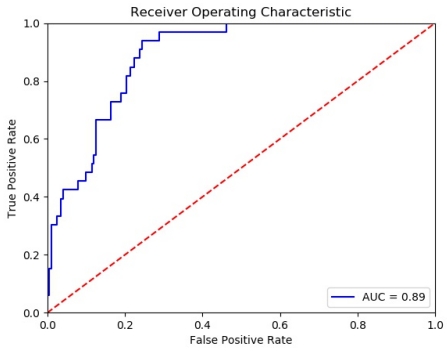
3.1.3.1 Method

Radiologists use a systematic approach in reading a chest radiograph, both frontal as well as lateral. In a chest radiograph depending on the darkness of the film, one can distinguish between the boundaries of soft organs (like lungs, trachea), hard organs like bones, and other artifacts like any outside body if injected by the patient. Black corresponds to air, dark grey corresponds to fat tissues, light grey corresponds to soft tissue, off white corresponds to bones, and bright white corresponds to metallic or outside bodies, it can also be a pacemaker in the patient's heart. Whenever there is a density change between two corresponding regions, there is a clear distinctive boundary that can be seen in the radiograph.

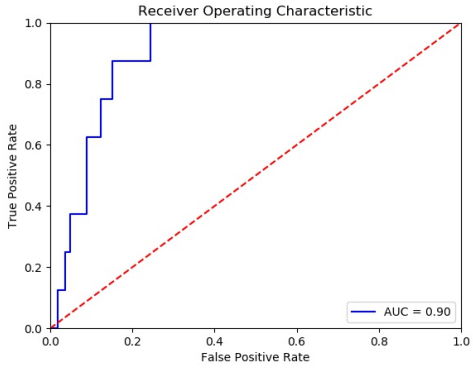
[2]https://github.com/abkds/chexpert



(a) Atelectasis



(b) Consolidation



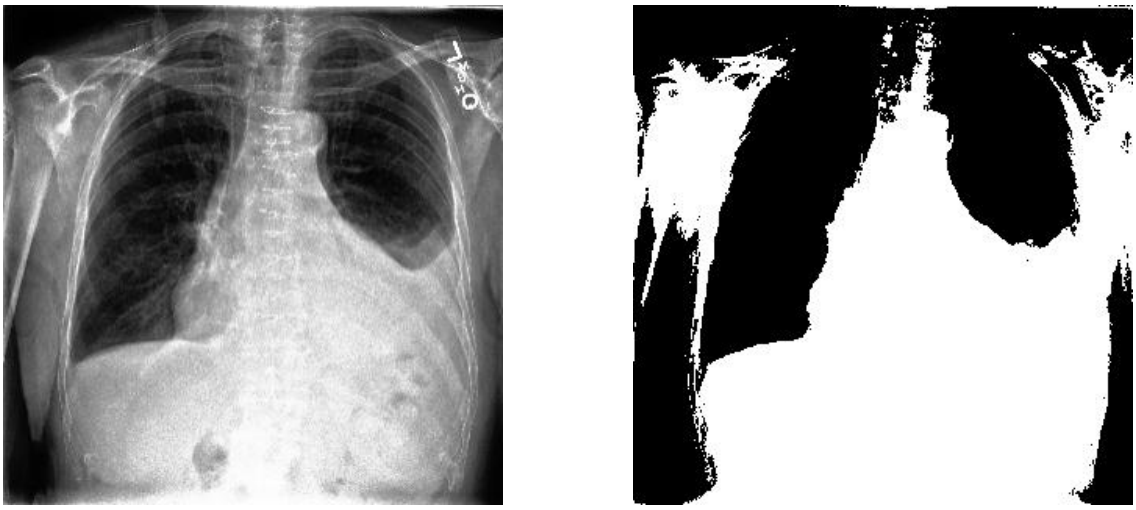
(c) Pneumonia

Figure 3.3: ROC Curves for Attention-Densenet

Atelectasis is a disease which is most easily identifiable from the radiograph. In this disease, one lung deflates or shrinks compared to other organ and there is significant breathing problem for the patient. We believe that, even though the densenet attention model provided a ROC-AUC score of 0.789, it can definitely do much better. Densenet model might be getting bogged down by the various other artifacts in the radiograph whereas the most important thing to concentrate for finding Atelectasis is the size of the lungs itself. We feel that segmenting the image for the shape of lungs in Atelectasis provides a easy way to distinguish between the area of each lung. If one lung is substantially deflated in size compared to the other then it can be said that it is a case of Atelectasis.

We use Double-DIP [23] to do the image segmentation. The Double-DIP method is a state-of-the-art image segmentation technique based on coupled "Deep Image Prior" networks. It learns

to segment the image in an unsupervised way except for the image itself. Including the Double-DIP technique of segmenting the image in the pipeline for training the deep learning model can improve the performance of the model as it removes all the unwanted artifacts from the radiograph and generates a simple mask of the lung images. We show a sample generated mask in figure 3.4.



(a) Actual (b) Mask

Figure 3.4: Left: A sample radiograph of patient suffering from Atelectasis, Right: Image segmentation of the radiograph showing different lung sizes

3.1.3.2 Training and Results

For training the network with segmented images, our pipeline has now become slightly more complicated as we use an image segmentation step before feeding it to a neural network. Segmenting a single image of size 320×320 is very computationally expensive as the Double-DIP network takes about 8-10 minutes to get a single image. It is not feasible to take all the two hundred thousand images and do image segmentation on them. A very crucial detail to notice is that the shape of the lungs is conspicuous in frontal radiograph images compared to lateral radiograph images. So we decided to randomly sample 8000 frontal radiograph images from the dataset and segment those images. Out of 8000 images, we sampled 4000 images with atelectasis and 4000 with no

atelectasis. To save computational time, we resized the images to 80×80. Each image took about 3-4 minutes to be segmented. We ran 8 parallel jobs over a few days.

After doing the segmentation, we had a new dataset of masks. We used this dataset to train a new neural network specific to atelectasis. We went from a few simple models to more complex models to test, which performs better. A straightforward convolutional neural network was trained over the dataset to do the binary classification. The feed-forward net consisted of three convolutional layers with a filter size of 5×5 and a filter depth of 10, 20, and 20. We added a dropout layer after the first two layers and the third layer. There was a max-pooling layer after each convolutional layer, which was followed by the dropout layer. After that, we had a flatten layer. The last two layers were fully connected layers going from 720 to 50 and then 50 to 1 (since we are doing binary classification). Training this simple network on the 8000 images yielded a ROC-AUC score of 0.806, which already performed better than the previous atelectasis results. However, since we had very few images, we trained it again on densenet121 for 6 epochs. The best checkpoint in densenet121 gave a result of 0.885. This result was significantly better than all the previous results for atelectasis. We found that this technique of segmenting the images for the network increases the ROC-AUC scores for atelectasis. The ROC-AUC curve for atelectasis is shown in figure 3.5.

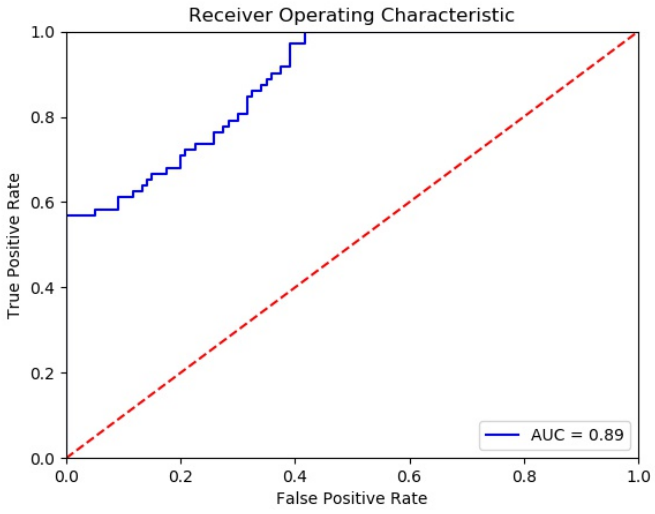


Figure 3.5: ROC Curve for Atelectasis after Image Segmentation

3.2 Clinical-Trial Matching

Our first approach is to segregate criteria which have basic conditions like $\geq$, $>$, $<$ and $\leq$. Criteria with the above comparisons can generally be viewed as structurally equivalent to a "Quantity Comparator Amount". The comparator is a set which comprises of $\{\geq, >, <, \leq\}$. First, we normalize all the criteria by replacing $\geq$ with "greater than or equal to", $\leq$ with "less than or equal to", $>$ with "greater than" and $<$ with "less than". Doing the normalization operation brings uniformity among all the criteria as it removes any symbols. In the next phase, we segregate the criteria by using the longest common subsequence method. We match each criterion against the four phrases discussed previously; it is considered to be a match if the longest common subsequence length comes out to be equal to the phrase length. We do the segregation in the order of the longest phrase to the smallest phrase because the smallest phrase in our case is a part of the longer phrase.

Following are some representative examples of criteria after doing the segregation.

- **less than**
  - Geriatric Depression Scale (GDS) score of less than 6;
  - a score of less than 10 points on the HAM-D-17
- **greater than**
  - Alcohol intake greater than 30 grams (drink more than 2 beers OR equivalent per day).
  - Mini Mental State Examination score greater than 15 / 30
- **greater than or equal to**
  - Age greater than or equal to 55 years with a diagnosis of MCI
  - MMSE score greater than or equal to 24
- **less than or equal to**
  - Rosen-Modified Hachinski Ischemia Score less than or equal to 4.
  - Creatinine less than or equal to 2x institutional upper limits of normal

3.2.1 Variable Detection

Before jumping into any complicated machine learning models, it is essential to have a baseline for our task of variable detection. Each criterion has a particular medical term (or variable), which is compared against some numeric value to identify a particular patient. It is crucial to identify that variable as that will help in the pipeline further by helping our user of the patient database even if he/she is not well versed with the medical vocabulary. Furthermore, in the case of an automated process to generate the query, the same identified variable can be directly used to fill in the query column names.

For the baseline, we use a two-fold technique and use a similarity measure called the Jaccard Similarity Score. The Jaccard Similarity Score between two pieces of text is given by

$$J(X, Y) = \frac{|X \cap Y|}{|X \cup Y|}$$

here X and Y are both tokenized strings. For a given criterion, we first find the criteria similar to the given criteria by using the Jaccard similarity score and order them in decreasing order of similarity. Since medical term (or variable) is most likely the term that finds similar criteria, we then take the Longest Common Subsequence with the original criteria, and this common subsequence represents our medical term. Although this is not a fool-proof technique, it works reasonably well in finding the medical terms.

Following are some examples of variable detection:

- Criteria 1: Apathy Evaluation Scale-Clinician (AES-C) score of greater than or equal to 30
  Most likely variable is "apathy evaluation scale-clinician aes-c score" or "apathy evaluation score"

- Criteria 2: Mini-Mental State Examination (MMSE) greater than or equal to 25
  Most likely variable is "mini-mental state examination mmse".

- Criteria 3: Non-demented: Montreal Cognitive Assessment greater than 26 and Clinical

Dementia Rating = 0

Most likely variable is "montreal cognitive assessment clinical dementia rating" or "clinical dementia rating"

- Criteria 4: Serum creatinine greater than 1.5 times the upper limit of normal

  Most likely variable is "serum creatinine".

We also provide examples of text similarity with the similarity score in tables 3.4, 3.5, 3.6 and 3.7. Our baseline works pretty well for smaller criteria which have just one variable. For each of the criteria given below the top 6 similar criteria matched in decreasing order of similarity score has been presented.

Example 1: Body mass index greater than 22.5

Similarity Score	Criterion
0.75	body mass index less than 18.5
0.75	Normal body mass index (BMI)
0.75	Body mass index of 18-32.
0.6	Body mass index between 18-32kg/m2
0.6	Body mass index (BMI) greater than 40
0.6	Have a Body mass index (BMI) greater than or equal to 23

Table 3.4: Similar criteria to "Body mass index greater than 22.5"

Example 2: Has cancer or a malignant tumor, untreated thyroid disorder or has a history of seizure disorder

Example 3: Cerebral Spinal Fluid (CSF) Amyloid Beta 1-42 (A42) less than or equal to 600 pg/mL, or A ratio of total tau to A42 greater than or equal to 0.39.

Example 4:Clinical evidence or history of cerebrovascular accident; transient ischemic attack; significant head injury with associated loss of consciousness, skull fracture or persisting cognitive impairment; other unexplained or recurrent loss of consciousness for greater than or equal to 15 minutes

Similarity Score	Criterion
0.88	Subjects who have cancer or a malignant tumor, untreated thyroid disorder, or a history of seizure disorder.
0.375	History of a seizure disorder.
0.375	History of seizure disorder
0.333	History of stroke or seizure disorder
0.333	History of seizure disorder or epilepsy
0.333	History or presence of seizure disorder

Table 3.5: Similar criteria to "Has cancer or a malignant tumor, treated thyroid disorder or has a history of seizure disorder"

Similarity Score	Criterion
0.66	Cerebral Spinal Fluid (CSF) Amyloid Beta 1-42 (Aβ42) less than or equal to 600 pg/mL, or A ratio of total tau to Aβ42 greater than or equal to 0.39.
0.33	Cerebral spinal fluid (CSF) result consistent with the presence of amyloid pathology
0.2	The patient has a ratio of total tau/Aβ42 in cerebrospinal fluid greater than or equal to 0.28.
0.2	The patient has a ratio of total tau/Aβ42 in cerebrospinal fluid greater than or equal to 0.30.
0.2	The patient has a ratio of total tau/Aβ42 in cerebrospinal fluid greater than or equal to 0.30.
0.2	The patient has a ratio of total tau/Aβ42 in cerebrospinal fluid greater than or equal to 0.28.

Table 3.6: Similar criteria to "Cerebral Spinal Fluid (CSF) Amyloid Beta 1-42 (A42) less than or equal to 600 pg/mL, or A ratio of total tau to A42 greater than or equal to 0.39."

As we saw earlier that the method we used to find variables in particular criteria is miserable when it comes to fetching out all the variables in long criteria. Even without using any natural language processing methods, we were able to extract the variables pretty consistently for most of the cases. After finding the first match of the current criteria, we find the common words between both the criteria and store them in a list. For this list of common words, first, we generate another list, which is essentially mapping to the positions (or indices) of the common words into the list. After that, we find all the contiguous positional indices from this list. Finally, we combine each

Similarity Score	Criterion
0.52	Clinical evidence or history of stroke, transient ischemic attack, significant head injury or other unexplained or recurrent loss of consciousness greater than or equal to 15 minutes
0.290	History of head trauma or injury causing loss of consciousness, lasting more than three (3) minutes or associated with skull fracture or intercranial bleeding or abnormal MRI.
0.259	history of head trauma associated with injury-onset cognitive complaints or loss of consciousness for 10 minutes or longer.
0.25	History of severe head injury (with loss of consciousness greater than 30 minutes)
0.25	history of significant head trauma with loss if consciousness greater than 10 minutes
0.20	History of stroke or multiple (greater than 3 discreet episodes) Transient Ischemic Attacks (TIAs), severe head trauma with cognitive sequelae, uncontrolled seizures, or unexplained prolonged loss of consciousness (greater than 1 minute) during the past year.

Table 3.7: Similar criteria to example 4

contiguous positional index list and fetch the name of the variable. As an example, let us take the following criteria: *Has cancer or a malignant tumor, untreated thyroid disorder or has a history of seizure disorder*. The first match for this criteria is *Subjects who have cancer or a malignant tumor, untreated thyroid disorder, or a history of seizure disorder*. The list of common words is as follows, ['cancer','malignant','tumor','untreated','thyroid','disorder','history','seizure','disorder']. Now we map this to an array of positional indices, which for this example will result in the following array [1, 4, 5, 7, 8, 9, 13, 15, 16]. Now from this list, we can see that the list of the positional indices which are contiguous are [[1], [4, 5], [7, 8, 9], [13], [15, 16]]. With this final list, we map it back to the name of the variables, which in this case results in ['cancer', 'malignant tumor', 'untreated thyroid disorder', 'history', 'seizure disorder']. We can see that our simple method was able to find all the four variables correctly while incorrectly finding *history* as well, which is certainly not a variable. We apply this technique to various criteria, some examples have been shown in table 3.8. We see that the recall is high as it identifies all the variables, but precision varies depending on the number of common words. The algorithm to find the list of contiguous indices from the list

of indices has been given in the appendix for reference A.1.

Criterion	Variables Detected
Has cancer or a malignant tumor, untreated thyroid disorder or has a history of seizure disorder	cancer, malignant tumor, untreated thyroid disorder , history, seizure disorder
Clinical evidence or history of cerebrovascular accident; transient ischemic attack; significant head injury with associated loss of consciousness, skull fracture or persisting cognitive impairment; other unexplained or recurrent loss of consciousness for greater than or equal to 15 minutes	Clinical evidence, history, cerebrovascular accident, transient ischemic attack, significant head injury, loss, consciousness, skull fracture, persisting cognitive impairment, unexplained, recurrent, consciousness, minutes
Must have at least two of the following: resting tremor, bradykinesia, rigidity (must have either resting tremor or bradykinesia); OR either asymmetric resting tremor or asymmetric bradykinesia	least two, resting tremor, rigidity,bradykinesia
Poorly controlled high blood pressure (systolic blood pressure of 160 mmHg or higher and/or diastolic blood pressure of 100 mmHg or higher) despite treatment during the 3 months prior to dosing, or treatment refractory high blood pressure, defined as treatment requiring 3 or more antihypertensives from different classes.	high blood pressure, systolic blood pressure, mmHg, higher and/or diastolic, blood pressure, mmHg, higher, despite treatment,months

Table 3.8: Examples of variable detection

3.2.2 NDCG

NDCG or Normalized Discounted Cumulative Gain helps us in measuring the ranking quality of a search algorithm. It uses a graded relevance scale. The primary assumption is that more relevant documents should be at the top of the returned list. "Gain" in this context means usefulness. NDCG comes from Discounted Cumulative Gain, which in turn comes from Cumulative Gain.

Cumulative Gain (CG) is the simply the sum of the relevances returned by the system. Let us assume that an information retrieval system returns p results and the document d_i has relevance

rel_i then the Cumulative Gain is given by

$$\mathrm{CG}_p = \sum_{i=1}^{p} rel_i$$

We can see from the definition that Cumulative Gain does not take into account the order of the retrieved documents. Even if the most relevant document is moved to a different position, then the Cumulative Gain will not change. We want a metric that gives more weightage to an important document being at the top of the list.

Discounted Cumulative Gain (DCG) builds of top of Cumulative Gain. It penalizes the relevance of a document retrieved at a lower rank. The penalizing factor is given $\log(i+1)$ where i is the rank of the document. DCG is exactly given by

$$\mathrm{DCG}_p = \sum_{i=1}^{p} \frac{rel_i}{\log(i+1)}$$

To compare the DCGs across query search results, which can be of varying lengths, we need to normalize the DCGs. The core idea is that if a search query returns p documents, we consider the best retrieved relevant p documents and calculate the Ideal DCG. Ideal DCG, as it sounds, is the best an information retrieval can do. Dividing DCG by IDCG gives is NDCG

$$\mathrm{nDCG}_p = \frac{\mathrm{DCG}_p}{\mathrm{IDCG}_p}$$

The NDCG value always lies between 0.0 and 1.0, making it easier to compare the different query search results. We judge a system by the average NDCGs across all the test queries.

We handpicked 24 data points in our case and marked the most relevant retrieved results manually. We use the mean NDCG score calculated across this dataset as a metric for our system.

3.3 Embeddings

3.3.1 BioSent2Vec

We use vanilla BioSent2Vec [16], which is a deep learning model based on the Sentence to Vector based model but has been trained on the MIMIC-III dataset. It generates a $\mathbb{R}^{700}$ size embedding for each sentence. Following is an example of the similarity scores generated and similar sentences. We use two previously used examples to show how the model performs in tables 3.9 and 3.10

Similarity Score	Criterion
0.84	Clinical evidence or history of stroke, transient ischemic attack, significant head injury or other unexplained or recurrent loss of consciousness greater than or equal to 15 minutes
0.82	History of major stroke, head injury with loss of consciousness of greater than 30 minutes, or other neurological/systemic illness that may affect cognition
0.809	History of stroke or multiple (greater than 3 discreet episodes) Transient Ischemic Attacks (TIAs), severe head trauma with cognitive sequelae, uncontrolled seizures, or unexplained prolonged loss of consciousness (greater than 1 minute) during the past year.
0.808	History of severe head injury (with loss of consciousness greater than 30 minutes)
0.79	history of head injury with loss of consciousness greater than 1 hour

Table 3.9: Similarity found using BioSent2Vec for example 4

Vanilla BioSent2Vec gives us an NDCG score of 0.4643

3.3.2 BERT Embeddings

The Natural Language Processing domain historically has suffered from the fact that there are not many massive labeled datasets for a variety of tasks like Question Answering and Sentiment Analysis. Even the vast datasets available have a maximum of a few hundred thousand data points. Modern deep neural networks require many data points to train and perform well on a required task.

Similarity Score	Criterion
0.94	Subjects who have cancer or a malignant tumor, untreated thyroid disorder, or a history of seizure disorder.
0.69	have a history of a seizure disorder;
0.66	History of seizure disorder or epilepsy
0.629	Seizure disorder, history of stroke, focal brain lesion, traumatic brain injury, substance abuse, malignancy.'
0.629	are totally blind in both eyes, have photosensitivity or photophobia, Parkinsons disease, known untreated sleep apnea or other sleep disorders, seizure disorder, bipolar, schizophrenia, are actively receiving chemotherapy or radiation therapy for cancer.
0.581	Personal medical history and/or clinically determined disorders: current B12 deficiency, positive syphilis serology, chronic sinusitis or any untreated thyroid disease, significant head trauma, or history of difficulty with smell and/or taste prior to AD diagnosis.
0.577	Examples include malignant cancer, chemotherapy, untreated thyroid disease, heart failure, or renal insufficiency.

Table 3.10: Similarity found using BioSent2Vec for example 2

In the domain of images, various transfer learning methods were pretty standard, which allowed researchers to start from a set of initial weights transferred from networks that have already been trained on enormous datasets like imagenet [4].

3.3.2.1 Transfer Learning

Transfer learning is the idea of exploiting knowledge learned from one task to solve related tasks. For most of its brief history, Neural Networks were operating in the domain of isolated learning, which meant training a neural network from scratch for solving a problem and training it again from scratch for any other task. Getting vast amounts of data like imagenet is not possible for a myriad of other tasks. True Artificial General Intelligence (AGI) can only be reached if the knowledge gained from other tasks is utilized on specialized tasks. Such a system can solve many problems and will not be bogged down by the fact that most machine learning algorithms require vast amounts of data.

Take, for example, humans rarely learn anything from scratch. Most of the time, all we do is

transfer skills learned from one domain and sharpen that skill pertaining to the new domain (fine-tuning in neural network literature). If we are learning statistics, learning machine learning is not that of a challenging task, and many skills are transferable. Humans also learn passively without directly participating. In the case of driving a car, even when we have not driven but sat with someone who drives a car, we have seen the movements. When the time comes to drive, humans fully utilize this prior knowledge to learn the skill quickly. This example serves as motivation that learning from a prior knowledge only saves time, but the knowledge helps in generalizing to a myriad of other tasks.

Imagenet correctly solved the problem of vast amounts of data for image tasks by providing millions of labeled images and thus ushering unparalleled innovation in the domain of neural network for images. The process of labeling a vast amount of images required years of painstaking hard work of researchers from Stanford. Since no other domain has such a vast labeled dataset, training a network on specific task results in a network which excels at that task but is hardly generalizable to any other task.

Classical machine learning means that you have two tasks Task 1 and Task 2 and two datasets, Dataset 1 and Dataset 2, but independent models are learned on each of them even when the tasks might be similar (not same). In the case of transfer learning, Task 1 is a task which can serve as general knowledge which can encompass broad knowledge by training on huge datasets and then the knowledge is transferred to another network whose task is to train a model on another relevant dataset.

There are three main questions in transfer learning: 1. What to transfer? Which part of knowledge is transferable from source learner to target keeping in mind which common knowledge would benefit the target most. 2. When to transfer? One needs to be cognizant of scenarios where transferring knowledge can worsen the performance. 3. How to transfer? Identifying ways in which the knowledge can be transferred, one of the most common ways being transferring the model weights to the new network.

This naturally brings us to BERT and many other BERT derivatives, which have been state-

of-the-art on a variety of Natural Language Processing tasks. BERT revolutionized the domain by introducing transfer learning of an unparalleled scale. Transfer learning in NLP is not new; there were models like OpenAI-GPT, Infersent [14], Universal Sentence Decoder [15] but none of them were as good as BERT [18] in generalizing to new tasks and at the same time easily adaptable to new tasks. BERT built on the idea of transformers, which are its core module stacked on top of each other. Transformers were first introduced by Vaswani et al. [17].

3.3.2.2 BERT Structure

BERT is deeply bidirectional, unsupervised language representation pretrained on a plain text corpus. "Deeply bidirectional" gave it an edge over other such models of the time, and being unsupervised gave it access to limitless text corpora. The "deeply bidirectional" property made BERT very powerful. BERT is made of stacking transformers on top of each other; for the small model, they stacked 12 and 24 for the bigger one. The key ingredient which made the model shine was not just the bidirectional model but also the tasks on which it was trained. Training on language modeling and using the bidirectional model is like cheating the system since the model already has access to future words, which it obviously does not have during inference time. The authors of BERT came up with Masked Language Model and Sentence Entailment tasks. In Masked Language Model, a random small percentage of words is omitted from a sentence, and those words are made to predict. In the Sentence Entailment task, BERT is made to predict if a sentence B naturally occurs after A.

3.3.2.3 BioBERT

Our task pertains to the biomedical domain. It is a known fact in computer science that for any algorithm or neural network model that "Garbage in, garbage out". Using BERT trained on the Wikipedia corpus or book corpus for tasks on biomedical texts will result in inferior results as the text in the biomedical domain is very different from everyday speech and includes dense jargon and domain-specific terms. Researchers trained the BERT model further on the PubMed and MIMIC-III datasets; they fine-tuned the model for 23 days resulting in weights which they call BioBERT

[24]. In essence, BioBERT is just BERT trained on a biomedical text corpus. BioBERT uses WordPiece [25] tokenization; it mitigates the out-of-vocabulary issue. With the help of word piece tokenization, any new word can be broken down into smaller, more frequent subwords. Instead of having WordPiece tokens specific for the biomedical domain, the authors decided to continue with the BERT WordPiece tokens to have compatibility with BERT. The authors also point out that cased tokens give slightly better results on tasks compared to uncased ones, which is to be expected since biomedical texts generally contain terms which are cased.

3.3.2.4 Sentence BERT

Our goal is to find sentence similarity between by using similarity between pairs of sentences. This process poses an immediate challenge in a model like BERT because even for a dataset of $n = 10000$ sentences, we have a total of $\frac{n \cdot (n-1)}{2}$ pairs which are equal to 49995000. At inference time, let us say we want to find the pair with the highest similarity; it will take a considerable amount of time and make the computation infeasible, and with the quadratic relation, even small dataset sizes are already in the range of 50 million. To cope up with this, Reimers et al. [26] proposed Sentence-BERT or SBERT (figure 3.6). They used the classic Siamese twin network technique to decouple the networks so that we have two networks which are sharing the same weights. This way, the same network is independently used to generate the embeddings of each sentence in a pair. BERT produces various embeddings as output, `[CLS]`, and output vectors for each token in the sentence. They use three strategies for generating an embedding for sentence similarity tasks. First is using the `[CLS]` itself; second is pooling the output vectors and taking a mean which is referred to as `MEAN` and the third is taking a max of all the output vectors called the `MAX` strategy.

The authors show that BERT does not readily capture the meaning of a sentence in the embedding produced at the `[CLS]` token. The hypothesis is that BERT does not train on language modeling tasks, instead it trains on Masked Language Model which does not necessarily give better sentence embeddings. However, since BERT already has much knowledge embedded, it is easy to fine-tune it on an NLI dataset to generate useful sentence embeddings. So much so, their network

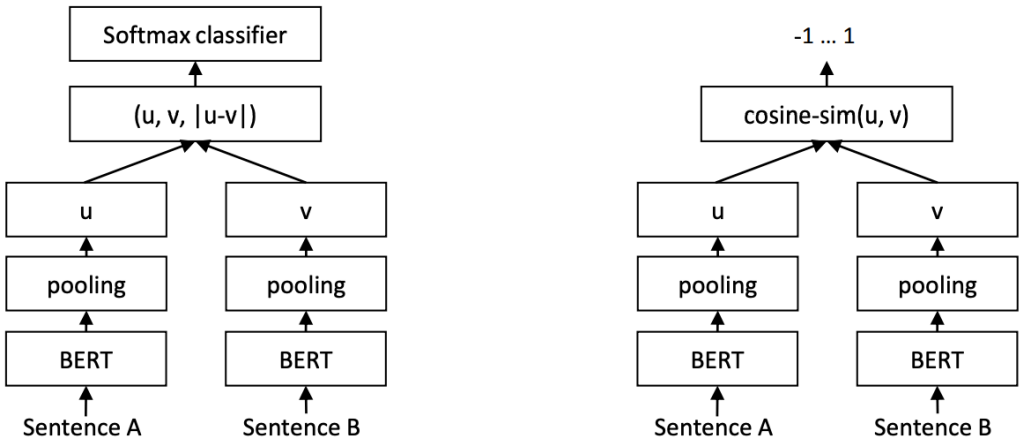


Figure 3.6: Sentence BERT Architecture [3]

outperformed the state-of-the-art at that time with fine-tuning on the SNLI dataset.

3.3.3 Methodology

With the knowledge that BERT performs well, we use BERT for sentence similarity in our task. As mentioned earlier, it did not make sense to start with original BERT weights (trained on WikiCorpus) for our biomedical domain; we directly went ahead and started with BioBERT weights, which were published by the authors. BioBERT did not necessarily generate the correct sentence embeddings needed for the sentence similarity task. It was imperative to train it on a Natural Language Inference task to train it to encode the meaning of a biomedical sentence accurately. The most appropriate NLI dataset for the same was MedNLI [27]. MedNLI contains entailment information on medical texts, where each data point contains a pair of sentences. A data point has the basic structure of (S_A, S_B, y) where S_A is the first sentence, S_B is the second sentence, and y is the label. The label y can be either entailment, neutral, or contradiction depending on the sentence S_B. The MedNLI dataset contains a total of $11,232$ points in the training set and a total

[3]Reprinted with permission from "Sentence-bert: Sentence embeddings using siamese bert networks" by N. Reimers and I. Gurevych, 2019 Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, 3982-3992, Copyright 2019 by Association for Computational Linguistics.

of $1,395$ sentences in the dataset. We fine-tune the SBERT network loaded with BioBERT weights for four epochs.

We use a custom dataset which comprises of clinical trial texts for Alzheimer's disease patients. We bootstrap the dataset by using BioSent2Vec [16]. BioSent2Vec is an earlier state of the art model which generates embeddings for medical text. We use the embeddings generated from this dataset and apply k-Nearest Neighbors to the embeddings. K-Nearest Neighbors generates a bunch of clusters that we manually label to generate 1000 data points with positive and negative pairs showing which sentences are similar and which are not.

From here onward, we refer to SBERT with BioBERT weights as SBioBERT. SBioBERT trained on the MedNLI evaluates to the results in table 3.11. The evaluation shows that there is ample scope for improvement in the metrics. Before continuing the training on our dataset, we changed the head of SBioBERT and replaced it with a multi-layer deep neural network with a regressive loss function. We take the embeddings from the first sentence and generate the embedding u. From the other sentence, we generate the embedding v; we take concatenation of $|u-v|, u, v$ as the input to the deep neural network head and in the generate an activation between 0 and 1, we use Pearson loss to optimize our network and then we further evaluate the metrics on our dataset which have been mentioned in table 3.12.

Cosine Similarity	
Pearson	0.5315
Spearman	0.6791

(a)

Manhattan Distance	
Pearson	0.6564
Spearman	0.6807

(b)

Euclidean Distance	
Pearson	0.6423
Spearman	0.6763

(c)

Dot Product	
Pearson	0.4293
Spearman	0.4831

(d)

Table 3.11: Sentence Similarity Metrics on the development dataset after training on MedNLI

SBioBERT trained on MedNLI returns an NDCG score of 0.4736 on our system. Continuing training the SBioBERT on our custom similarity dataset improves the NDCG score from 0.4736 to 0.6210

Cosine Similarity	
Pearson	0.9366
Spearman	0.9196

(a)

Manhattan Distance	
Pearson	0.9173
Spearman	0.8903

(b)

Euclidean Distance	
Pearson	0.9173
Spearman	0.8904

(c)

Dot Product	
Pearson	0.8468
Spearman	0.8643

(d)

Table 3.12: Sentence Similarity Metrics on the development dataset after training on MedNLI and continuing training on our dataset

3.3.4 Named Entity Recognition

Most of the medical texts (trial texts in our case) revolve around certain conditions or diseases, which are the "core" topics being discussed in a piece of trial text. Trials have mentions of lots of these diseases in a single text. Let us take a straightforward and concrete example of a text. In the example text, 'Has a malignant tumor, untreated thyroid disorder or has a history of seizure disorder', we see that there are three topics which have been underlined. Let us say this is the query, and we are trying to rank the documents, and we use a model like BERT or any other model which tries to encode the entire sentence into a single embedding. The model tries to encode as much information as possible in one array of numbers, which has to take into account all the topics mentioned. In the medical domain, literature shows that it is better to match against particular topics, if they are present, to keep the recall high. In such a scenario, one would try to do a simple text match, but for that, we need to know the named entities.

A model like SciBERT [28] can help us in identifying the set of named entities that are going to be the core topics of a query text. Let us take an example query which contains the phrase "seizure disorder". The NER model will identify this phrase as an entity. It is possible that in the documents, there are terms like "epileptic disorder". Instead of doing naive text-matching, we can first find the distance between the embeddings of both the phrases found by the NER model. Whenever the match is high, that is a probable candidate. This process will weed out many non-candidates and leave us with a small set of candidates on which we can run powerful BERT (or any other model) to help us with the ranking.

We combine classic, and machine learning approaches to build a robust entity detector. We take

the weights made publicly available by the authors of SciBERT and BioBERT and train them on two medical entity detection datasets NCBI-Disease [4] and bc5cdr [5]. The F1-scores for the training on both the datasets have been presented in table 3.13. BioBERT on bc5cdr performs the best with an F1-score of 0.8665. We use the BioBERT model for named entity recognition. One caveat is that in the entity recognition datasets, the only label available is `ENTITY` without distinguishing between the different types of medical entities.

	SciBERT	BioBERT
NCBI-Disease	0.8499	0.8324
bc5cdr	0.8621	**0.8665**

Table 3.13: Model performance on Medical NER datasets

Our entity detector uses a two-pronged approach. We use a publicly available dataset PubMed Phrases [6]. PubMed Phrases comprises of coherent text segments that are beneficial for information retrieval and human comprehension. The dataset contains around $700,000$ phrases. Although huge, still the dataset is not exhaustive. Few examples of medical text which we were not able to find in the PubMed phrases are "trifascicular block", "dabigatran rivaroxaban", "apixaban". The longest n-gram in the PubMed phrases consists of 11 words, and the shortest n-gram consists of 2 words. The distribution of n-grams based on the number of tokens is given in table 3.14.

To make an efficient entity detector out of PubMed Phrases was not straightforward. Let us say we want to find the phrase "mental disorders". Doing a linear search over the dataset on 128gb memory and 48 core machine takes around 5 seconds for 100 searches. One might think of various ways to improve the performance of the search. The obvious next step is to hash the phrases, which does speed it up by a factor of 25000, but now we lose the ability to search if our phrase is a substring of a longer phrase. The solution to this problem was to build a suffix-trie data structure

[4]https://www.ncbi.nlm.nih.gov/research/bionlp/Data/disease/
[5]https://biocreative.bioinformatics.udel.edu/tasks/biocreative-v/track-3-cdr/
[6]https://www.ncbi.nlm.nih.gov/CBBresearch/Wilbur/IRET/DATASET/

n-gram	Phrase Count
1	0
2	373723
3	247577
4	66886
5	14260
6	2783
7	564
8	92
9	25
10	3
11	2

Table 3.14: Distribution of n-grams in PubMed Phrases

where the string was tokenized by breaking at spaces. Since the suffix-trie consists of the suffixes of the phrases, it increases the number of entries to $912,983$; searches in trie are 250 times faster than the linear search.

Our final algorithm first predicts the named entities using the machine learning models and then uses the trie based data structure to find the entities. We go for the longest possible n-gram by searching in the reverse order of the generated n-grams from the text starting from all the 11-grams to all 2-grams sequentially. As there can be repetitions between the phrases generated by both the algorithms or the entity generated by one algorithm may be a substring of another entity. We resolve such discrepancies and generate a final list of entities in the linear order of occurrence in the original medical text. Resolving the discrepancies, looks simple but has tricky edge cases. We provide the code for that in the appendix A.2

3.3.5 BM25 + SBioBERT

3.3.5.1 BM25

BM25 or Best Matching 25 is an information retrieval ranking algorithm which has its roots in the domain of probabilistic information retrieval. It was developed by Stephen E. Robertson, Karen Spärck Jones, and others in the 1970s and 1980s. BM25 improves on earlier information

retrieval methods, which consider term frequency (TF) and inverse document frequency (IDF). In a nutshell, term frequency means how many times a query occurs in a document, and inverse document frequency means the inverse of the frequency of term occurring in various documents. If a term is rare, then its inverse document frequency will be high, and hence it is given more weightage. Similarly, if a term occurs more in a text, then the document is about that term and should be given more weightage.

But we can see that there are drawbacks in just considering TF as it is. Let us say if one document has a term dog occurring 10 times than another document that does not necessarily mean that that document is 10 times more critical. BM25 introduces parameters to dampen the effect of term frequency. It quickly saturates the weightage given to term frequency even if the term frequency increases a lot. BM25 also takes into consideration the length of the document. If a document is small in length, then we can be very confident with more matches, but if the document is long, it takes time to be more confident of the matches. Combining everything we get BM25 with some nudgeable parameters as follows:

$$\text{score}(D, Q) = \sum_{i=1}^{n} \text{IDF}\left(q_i\right) \cdot \frac{f\left(q_i, D\right) \cdot \left(k_1 + 1\right)}{f\left(q_i, D\right) + k_1 \cdot \left(1 - b + b \cdot \frac{|D|}{\text{avgdl}}\right)}$$

Here k_1 is a hyperparameter which is generally set 1.25, D is the document, Q is the query. avgdl is the average document length, and $|D|$ is the document length of the current document at hand. Research has shown that BM25 can perform well on new corpuses comparatively better than deep learning networks, which have millions of parameters and can be easily affected by slight differences in the test distribution. Deep-learning based information retrieval is further affected by adversarial inputs [29].

3.3.5.2 Method

We extend the BM25 by incorporating BERT embeddings. Our idea at its core is pretty simple. The original BM25 depends on the exact phrase or text matches to identify relevant text. Extract phrase matching seems too harsh. For example, if we have we have to match a word like "disease",

"disorder" would also be a relevant match but BM25 cannot possibly identify that. We build the embeddings by using BM25 weights and individual tokens instead of taking the BERT embeddings directly; this will give more weight to tokens, according to BM25. We do the same thing with the query before applying to match the documents.

Searching through an array of vectors can be very slow. For this, we used a vector index library called Faiss [30]. Faiss is a vector index building software that helps in efficient searches based on similarity metrics like Cosine similarity or Euclidean distance. Faiss also provides GPU support. We index our dataset and create a flat index of vectors based on BM25 embeddings. Whenever we get a query, we create a BM25 based embedding on the fly and find the nearest neighbor by using Faiss search methods and return the top k matches. This method provides substantial increase over all the previous methods giving us a NDCG score of 0.8070. A compilation of all the NDCG scores is given in table 3.15.

Model	NDCG
BioSent2Vec	0.4643
SBioBERT + MedNLI	0.4736
SBioBERT + MedNLI + Sentence Similarity	0.6210
BM25 + SBioBERT	**0.8070**

Table 3.15: NDCG Scores for various models

3.3.6 TextMed - A visual tool for analyzing medical text

To complement our work we made a visual tool for analyzing these medical criteria. We provide the end user a smooth interface to enter text and search for similar text and entities in text. The interface will help the user in identifying the core concepts in a medical text.

In figure 3.7, we see that component 1 is the input search box where the user enters the input for querying. The user can press the yellow color button to analyze the text whenever the user is ready. Component 4 gives the user a choice to select a model from a list of pretrained models.

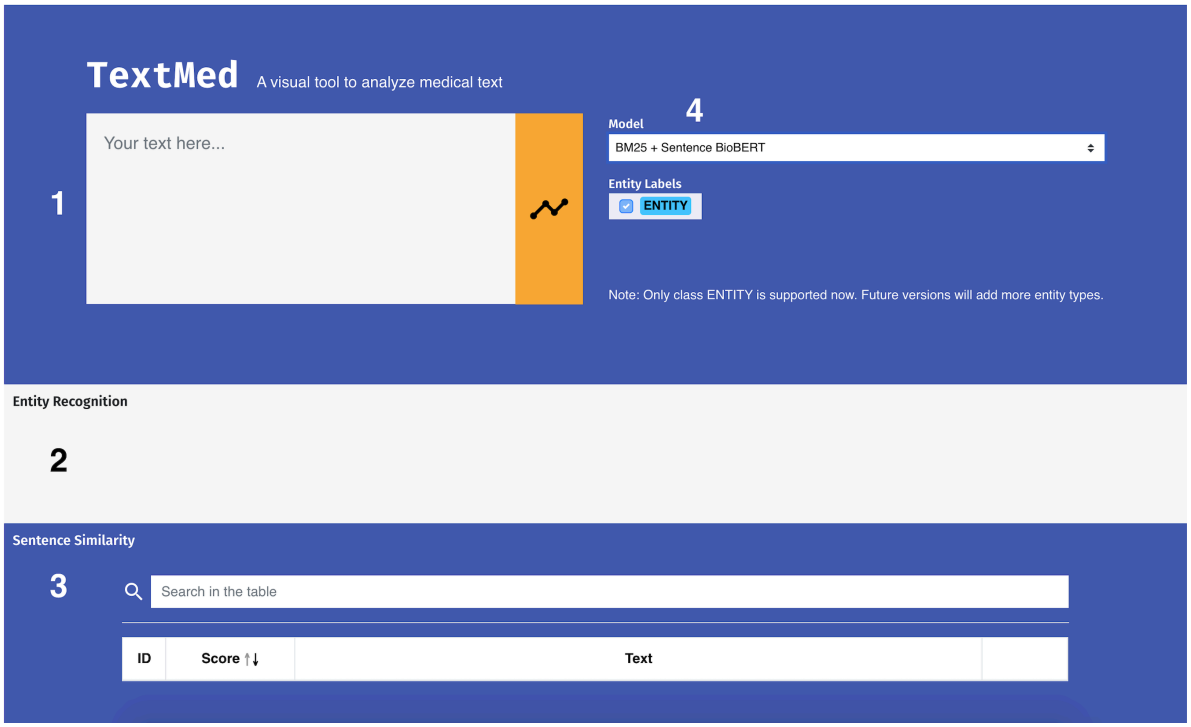


Figure 3.7: TextMed tool - Interface

Currently, the user can only choose `ENTITY`, which the user wants us to identify in the text. Later versions will support named entities, and the user can choose from them. After the user presses the search button, the server will return an analysis of text consisting of the entities in the text and the most similar texts. Component 2 will display the entities in the text. Component 3 will display the table which contains the top five similar texts in decreasing order of similarity. The table provides an option to search over the retrieved criteria. The user can also download the retrieved criteria in the form of a comma-separated file.

We use the criteria "History of stroke, transient ischemic attack, traumatic brain injury or severe cerebrovascular disease by clinical diagnosis or past MRI/CT; " as an example.

The Entity Recognition component show in figure 3.8 displays the entities identified by our algorithm. It boxes the entities and labels them. The entities for the current example are "History of", "stroke", "transient ischemic attack", "traumatic brain injury", and "cerebrovascular disease". Our model didn't identify MRI/CT as an entity in this example.

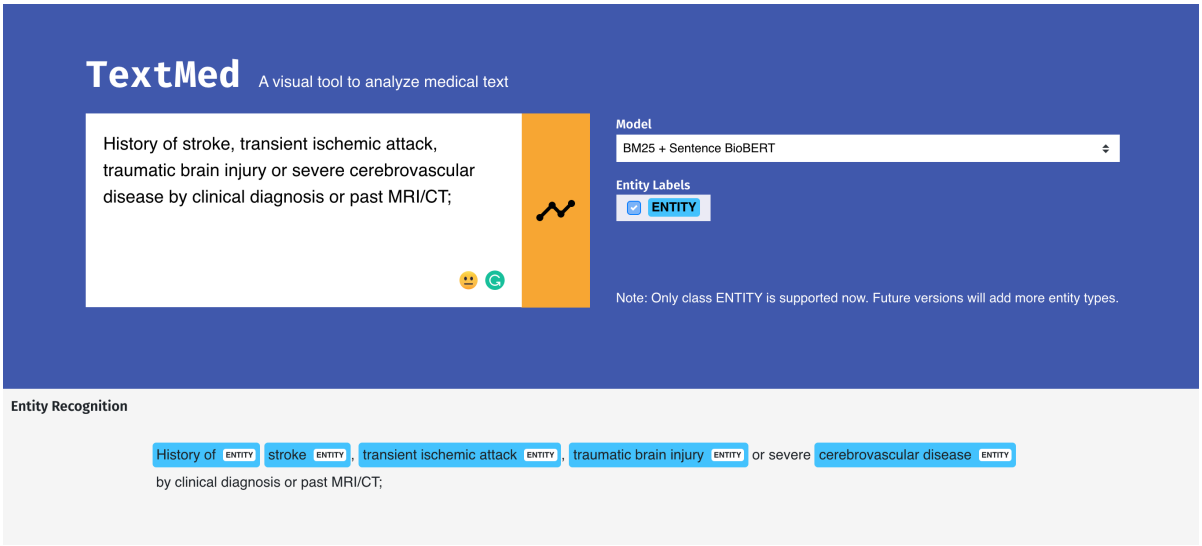


Figure 3.8: TextMed tool - Entity Recognition

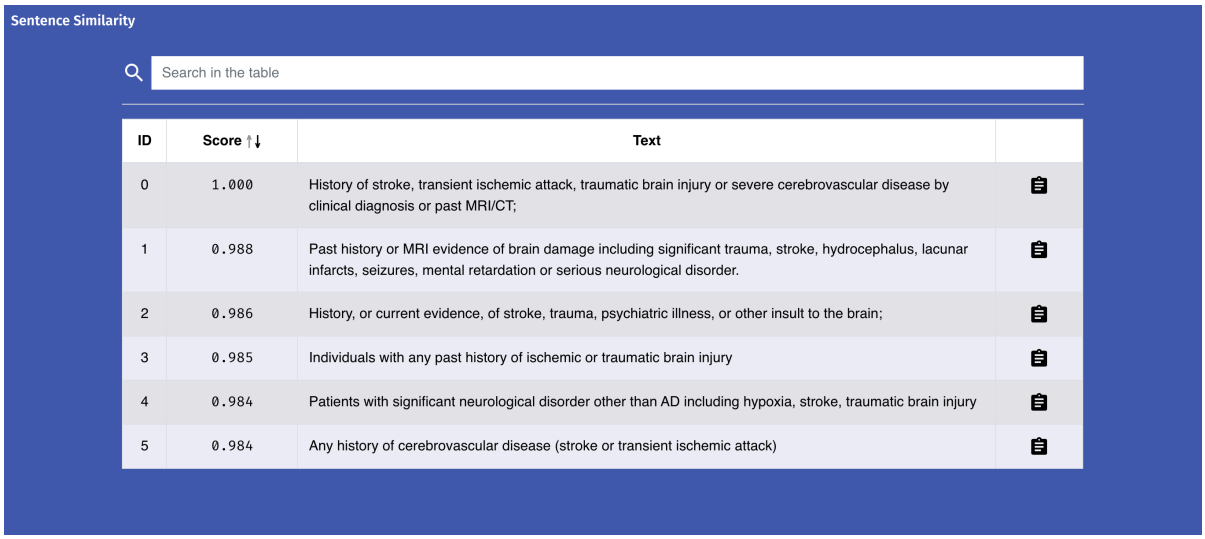


Figure 3.9: TextMed tool - Table

The table in figure 3.9 shows the retrieved similar medical texts. The first one is the query itself and the remaining are presented in order of decreasing similarity. The user can search through the retrieved text. We show the search functionality in figure 3.10. A search for the word "evidence" filters out the criteria that contain the word "evidence".

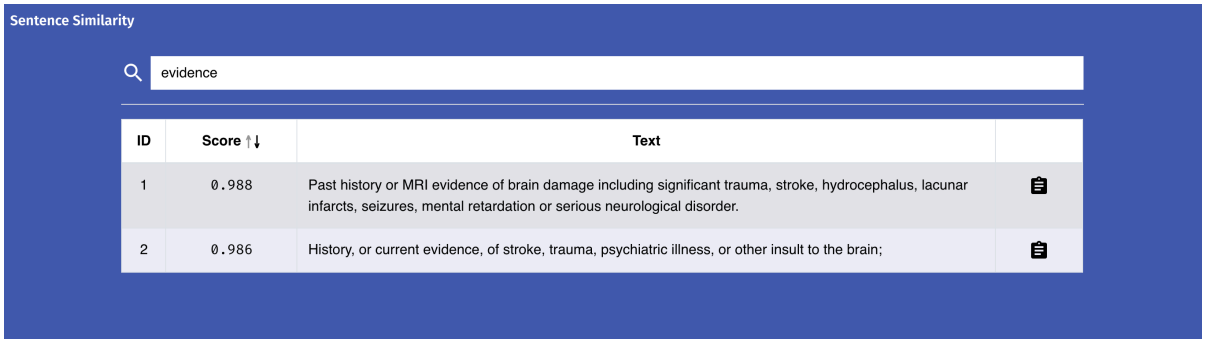


Figure 3.10: TextMed tool - Search

3.3.7 Fake Electronic Health Records

There are various legal, privacy, and security concerns in dealing with real Electronic Health Records. Anonymized EHRs are bought and sold by various institutions ranging from government, insurance and clinical groups [31]. Anonymized EHRs are also at risk for re-identification which can cause issues of privacy, consent, and confidentiality. It is imperative that to conduct research on EHRs, one needs some sort of real-looking fake data. It is impossible to get hold of real or anonymized EHRs.

Synthea [32], is an open-source software package that simulates the lifespans of synthetic patients, modeling the 10 most frequent reasons for primary care encounters and the 10 chronic conditions with the highest morbidity in the United States. We use Synthea in our workflow to extend our algorithms to actual patient matching.

3.3.7.1 Method

We generate a population of 5000 fake EHRs. Since our original criteria dataset revolved around Alzheimer's disease, our methods revolve around identifying patients with Alzheimer's like conditions. Out of 5000 patients, 228 have Alzheimer's disorder of varying degrees. Each patient generated by the Synthea tool contains certain essential conditions, observations over a period, patient details (name and address), and immunizations throughout the years. Synthea generates only past 10-year records of all the above.

The data is structured. Each condition the patient has, has a unique code referring to a medical database for which the resource urn is also given. However, since criteria are free text, this makes the entire problem interesting. Let us say someone has "hypothyroidism," but the criteria might say "has a thyroid disorder", we still have to match "thyroid disorder" against hypothyroidism.

We develop a reverse index of terms occurring in the conditions and observations and then use the BM25 + SBioBERT model to generate embeddings for all the terms. When a query criterion comes, we first breakdown the query criteria into core terms and then match the individual core term with vectors in the vector index. If there is a match, we fetch the list of patient ids associated with that vector. We take the intersection or union of the patient identification numbers depending upon the structure of the sentence. If we are not able to fetch any core terms in the input criteria, we simply embed the entire criteria text and match it against the patient conditions and observations. Using this technique, we can fetch the relevant patients.

4. SUMMARY AND CONCLUSIONS

4.1 Medical Diagnosis

We apply progressively simpler to advanced techniques in identifying diseases in chest radiographs. We first apply simple augmentation techniques in hopes of increasing the size of the dataset. Modern deep neural networks require millions of data inputs. Our augmentation methods provide a way of artificially increasing the size of the data and balancing the dataset for proper training. Augmentation helps us in achieving better ROC-AUC scores over the baseline.

We come up with a novel architecture incorporating attention module in the densenet naming it attention-densenet. We hypothesized that many diseases have multiple artifacts in an image responsible for the diagnosis of the disease. Our attention module is inspired by the attention module used in NLP transformers. We show that our attention-densenet works appropriately for some pathologies by improving over the baseline. However, it fails to do so in other pathologies. We conclude that for the diseases in which it does not perform well, there are not many things to attend on, and the increase in the number of parameters causes overfitting, thereby harming the performance on the pertaining diseases.

We show that knowing the essential criteria of how doctors identify a particular pathology can significantly reduce the computation time and increase the efficiency of the machine learning models. Doctors identify atelectasis based on the difference in the size of the lungs, as seen in the radiograph. We showed that removing many unnecessary artifacts and just concentrating on the size of the lungs by using a mask improves the overall performance of the system. We conclude that even in the era of deep learning, knowing about the problem statement and using information about the domain can significantly help in improving the system performance.

Deep learning can solve a task if massive amounts of data are available. In certain domains like medical imaging, generating big datasets is challenging. CheXpert, the dataset we have used, contains around two hundred thousand data-points, which is modest for modern deep learning

algorithms. We show that a mix of novel architecture, augmentation, and knowledge about the domain can help us in making efficient systems.

4.2 Clinical-Trial Matching

We start with a straightforward baseline method that uses the Jaccard similarity measure to match texts. Jaccard similarity considers the exact phrase matching between two pieces of text without taking order into account. We use the phrase matching intuitively to retrieve variable names. We use stopwords to weed out matches that are not variable names. Although simple, this algorithm runs in $\mathcal{O}(n^2)$, where n is the number of sentences in the dataset. Exact phrase matching does not take into account variations in phrases which can be readily handled by machine learning algorithms.

We then move on to sentence embedding methods. In sentence embedding methods, the goal is to project the text into a $\mathbb{R}^d$ space where d is the number of dimensions and use a similarity metric to find the closeness. We start with a medical equivalent of sent2vec called BioSent2Vec. BioSent2Vec is trained on massive medical corpora. BioSent2Vec projects the sentence into a $\mathbb{R}^d$, and we use a cosine similarity metric to produce a ranking of similar texts.

BioSent2Vec is a massive model which is almost 21gb on disk. It is difficult to fine-tune such large models with small datasets. Our dataset contains merely 16k sentences. We then move on to BERT. BERT models are relatively smaller and can easily be fine-tuned with smaller datasets to get the desired results. The authors of BERT trained it on Wikipedia, which would be unsuitable for biomedical texts. We went ahead and used BioBERT, which is a BERT model trained on PubMed articles and the MIMIC-III dataset.

BERT models are not known for generating sentences which encompass the sentence semantics. We use Sentence BERT starting with BioBERT weights and train it on the MedNLI dataset. We then continue the training with a custom generated sentence similarity dataset and provide increasingly better results on the NDCG metric.

We hypothesize that medical texts generally revolve around a few core topics. Our first method of variable finding was relatively slow and not scalable to massive datasets. We build a named

entity recognizer based on the BioBERT model and trained it on the bc5cdr dataset. We couple this entity recognizer with a suffix-trie data structure based on the PubMed phrases dataset and build an efficient and accurate entity recognizer.

We combine the classical BM25 algorithm from information retrieval with the named entity recognizer trained from BM25, this model gives more weightage to terms specific to medical texts and can also handle variations in the phrase. We generate the embeddings for each document based on this method and store it in a fast vector index, which has been optimized for search. We use this method and achieve better NDCG scores than the previous models.

We combine all of our techniques and present them to the end-user in the form of a visual tool called TextMed. TextMed allows the user to enter a search query and returns a thorough analysis of the text. It displays all the entities present in the query text and returns the top 5 document matches. It allows the user to save the returned data in a comma-separated file. The user can also search in the table or sort the data if needed.

We use Synthea tool to generate fake patient data. We continue our research on patient matching. We use the BM25 + SBioBERT algorithm for generating the index vectors. We index the conditions and observations found in patient data. We build a reverse index for easy searching. We try to parse the original criterion and extract variables to find matches and then take union or intersection of the resultant sets to get the final set of patients satisfying a criteria.

4.3 Future Work

Since we have the inverted index of the patient with respect to observations and conditions, our next goal remains to correctly break down the criteria into a given set of conditions, which can then be used to do queries on the inverted index. We need to convert the criteria to a bunch of and or conditions. We can use classical NLP approaches like building the parse tree or newer ML approaches like those used for Sentence-to-SQL tasks. The ML approaches help in building SQL queries from natural language. These approaches can help us breaking the criterion into a set of conditions. Although the conditions exist in criteria as being present or not present, the observations generally have some value associated with it. We need to be able to parse the numeric

conditions correctly from the criterion so that it helps us in constructing the right query against the index.

We will construct a medical dataset equivalent to those of sequence to SQL tasks. Once we have such a dataset, we can think of training models or even apply classical algorithms to parse the criteria and have a benchmark for the generation of correct conditions. It is straightforward to map the conditions to the patients once they are generated accurately. Another challenge is to identify negation in the criterion. Many times it happens that the negation is not easy to identify from the sentence making the task more challenging. Our dataset should take into consideration negation examples to be able to generate useful queries.

We will integrate the feature for extracting conditions against the patient database into our TextMed tool. In this way, we will provide the user with a complete end-to-end experience.

REFERENCES

[1] A. Kaplan and M. Haenlein, "Siri, siri, in my hand: Whos the fairest in the land? on the interpretations, illustrations, and implications of artificial intelligence," *Business Horizons*, vol. 62, no. 1, pp. 15 – 25, 2019.

[2] L. Deng and D. Yu, "Deep learning: Methods and applications," *Foundations and Trendső in Signal Processing*, vol. 7, no. 34, pp. 197–387, 2014.

[3] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," in *Proceedings of the IEEE*, pp. 2278–2324, 1998.

[4] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[5] J. Irvin, P. Rajpurkar, M. Ko, Y. Yu, S. Ciurea-Ilcus, C. Chute, H. Marklund, B. Haghgoo, R. L. Ball, K. S. Shpanskaya, J. Seekins, D. A. Mong, S. S. Halabi, J. K. Sandberg, R. Jones, D. B. Larson, C. P. Langlotz, B. N. Patel, M. P. Lungren, and A. Y. Ng, "Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison," pp. 590–597, AAAI Press, 2019.

[6] D. M. Hansell, A. A. Bankier, H. MacMahon, T. C. McLoud, N. L. Müller, and J. Remy, "Fleischner society: Glossary of terms for thoracic imaging," *Radiology vol. 246,3*, pp. 697–722, 2008.

[7] Y. Ling, S. A. Hasan, M. Filannino, K. P. Buchan, K. Lee, J. Liu, W. Boag, D. Jin, Ö. Uzuner, K. Lee, V. Datla, A. Qadir, and O. Farri, "A hybrid approach to precision medicine-related biomedical article retrieval and clinical trial matching," in *TREC*, 2017.

[8] S. Robertson and H. Zaragoza, "The probabilistic relevance framework: Bm25 and beyond," *Found. Trends Inf. Retr.*, vol. 3, p. 333389, Apr. 2009.

[9] "Medical subject headings." https://www.nlm.nih.gov/mesh/meshhome.html".

[10] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems 26* (C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, eds.), pp. 3111–3119, Curran Associates, Inc., 2013.

[11] J. Chung, Çaglar Gülçehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *ArXiv*, vol. abs/1412.3555, 2014.

[12] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, p. 17351780, Nov. 1997.

[13] M. Pagliardini, P. Gupta, and M. Jaggi, "Unsupervised Learning of Sentence Embeddings using Compositional n-Gram Features," in *NAACL 2018 - Conference of the North American Chapter of the Association for Computational Linguistics*, 2018.

[14] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes, "Supervised learning of universal sentence representations from natural language inference data," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, (Copenhagen, Denmark), pp. 670–680, Association for Computational Linguistics, September 2017.

[15] D. M. Cer, Y. Yang, S. yi Kong, N. Hua, N. Limtiaco, R. S. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, Y.-H. Sung, B. Strope, and R. Kurzweil, "Universal sentence encoder," *ArXiv*, vol. abs/1803.11175, 2018.

[16] Q. Chen, Y. Peng, and Z. Lu, "Biosentvec: creating sentence embeddings for biomedical texts," *2019 IEEE International Conference on Healthcare Informatics (ICHI)*, pp. 1–5, 2018.

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *ArXiv*, vol. abs/1706.03762, 2017.

[18] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, (Minneapolis, Minnesota), pp. 4171–4186, Association for Computational Linguistics, June 2019.

[19] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman, "GLUE: A multi-task benchmark and analysis platform for natural language understanding," in *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, (Brussels, Belgium), pp. 353–355, Association for Computational Linguistics, Nov. 2018.

[20] M. D. Bloice, P. M. Roth, and A. Holzinger, "Biomedical image augmentation using Augmentor," *Bioinformatics*, vol. 35, pp. 4522–4524, 04 2019.

[21] G. Huang, Z. Liu, and K. Q. Weinberger, "Densely connected convolutional networks," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2261–2269, 2017.

[22] Z. Wang, N. Zou, D. Shen, and S. Ji, "Non-local u-nets for biomedical image segmentation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

[23] Y. Gandelsman, A. Shocher, and M. Irani, "double-dip: Unsupervised image decomposition via coupled deep-image-priors," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11018–11027, 2019.

[24] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang, "Biobert: a pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics*, 2020.

[25] M. Schuster and K. Nakajima, "Japanese and korean voice search," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5149–5152, 2012.

[26] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 11 2019.

[27] A. Romanov and C. Shivade, "Lessons from natural language inference in the clinical domain," 2018.

[28] I. Beltagy, K. Lo, and A. Cohan, "Scibert: Pretrained language model for scientific text," in *EMNLP*, 2019.

[29] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *CoRR*, vol. abs/1312.6199, 2014.

[30] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with gpus," *arXiv preprint arXiv:1702.08734*, 2017.

[31] L. Sweeney, A. Abu, and J. Winn, "Identifying participants in the personal genome project by name." 04/24/2013 2013.

[32] J. Walonoski, M. Kramer, J. Nichols, A. Quina, C. Moesel, D. Hall, C. Duffett, K. Dube, T. Gallagher, and S. McLachlan, "Synthea: An approach, method, and software mechanism for generating synthetic patients and the synthetic electronic health care record," *Journal of the American Medical Informatics Association*, vol. 25, pp. 230–238, 08 2017.

APPENDIX A

CODE

Listing A.1: Combine indices to extract variable

```
def find_contiguous(positions):
   i = 0
   contiguous = []
   while i < len(positions):
      current = [positions[i]]
      while i+1 < len(positions) and positions[i+1] == positions[i]+1:
         current.append(positions[i+1])
         i = i+1
      contiguous.append(current)
      i = i+1
   return contiguous
```

Listing A.2: Resolving entities generated from Machine Learning Model and PubMed Phrases

```
def all_start_indices(string, substring):
   string = string.lower()
   index = 0
   indices = []
   while index < len(string):
      found = _index(string, substring, index)
      if found is None:
         return indices
      else:
         indices.append(found)
         index = found + len(substring)
   return indices
```

```
def get_entities(data, verbose=True):
    orig = data.lower()
    data = clean_text(data)
    prediction = bc5cdr_NER.predict(data, verbose=False)
    entities = [' '.join(entity) for entity in combine_entities(prediction)]
    confirmed_entities, probable_entities = get_entities_from_pubmed(data)

    _entities = []
    for entity in entities:
        if len(entity.split()) > 4:
            _entities.extend(entity.split())
        elif entity == '':
            pass
        else:
            _entities.append(entity)

    if verbose:
        print("ML NER: ", _entities)
        print("Phrase ", confirmed_entities)
        print("Phrase ", probable_entities)

    entities_start = []

    all_entities = set(confirmed_entities).union(probable_entities).union(
        ↪ _entities)

    for entity in all_entities:
        entities_start.extend([(entity, idx) for idx in all_start_indices(orig,
            ↪ entity)])

    entities_start = sorted(entities_start, key=cmp_to_key(compare))
    return entities_start
```

```
def _uniq(entities_start):
   result = [('', -1)]
   for tup in entities_start:
      if tup[1] != result[-1][1]:
         result.append(tup)
   return result[1:]

def remove_substr(entities_start):
   if len(entities_start) == 0:
      return []
   result = [entities_start[0]]
   for tup in entities_start[1:]:
      if tup[1] > result[-1][1] + len(result[-1][0]):
         result.append(tup)
   return result

def _get_entities(data, verbose=False):
   return [e[0] for e in remove_substr(_uniq(get_entities(data, verbose=
      ↪ verbose)))]
```