B.TECH PROJECT REPORT

on

Disease Recognition, Summarized Medical Report Generation and Doctors Recommendation from X-Ray Images

By

Ashish Raj, 190003013

DISCIPLINE OF MECHANICAL ENGINEERING,
INDIAN INSTITUTE OF TECHNOLOGY INDORE

November 2022

Disease Recognition, Summarized Medical Report Generation and Doctors Recommendation from X-Ray Images

A PROJECT REPORT

Submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY
in
MECHANICAL ENGINEERING

Submitted by:
Ashish Raj,190003013

Discipline of Mechanical Engineering,
Indian Institute of Technology Indore

Guided by:

Dr. Chandresh Kumar Maurya,

Assistant Professor,

Computer Science and Engineering,
IIT Indore

INDIAN INSTITUTE OF TECHNOLOGY INDORE

November 2022

Candidate's Declaration

We hereby declare that the project entitled **"Disease Recognition, Summarized Medical Report Generation and Doctors Recommendation from X-Ray Images "** submitted in partial fulfilment for the award of the degree of Bachelor of Technology in Mechanical Engineering completed under the supervision of **Dr. Chandresh Kumar Maurya, Assistant Professor, Computer Science and Engineering,** IIT Indore is an authentic work.

Further, we declare that we have not submitted this work for the award of any other degree elsewhere.

Ashish Raj

Certificate by BTP Guide

It is certified that the above statement made by the student is correct to the best of my/our knowledge.

Dr. Chandresh Kumar Maurya
Assistant Professor
Discipline of Computer Science and Engineering
IIT Indore

Preface

This report on **“Disease Recognition, Summarized Medical Report Generation and Doctors Recommendation from X-Ray Images "** is prepared under the guidance of Dr. Chandresh Kumar Maurya, Assistant Professor, Computer Science and Engineering, IIT Indore.

Through this report, We have tried to provide a detailed description of our approach, design, and implementation of the method to address the problem of disease recognition, doctors recommendation, and summarization task of medical reports from x-ray images by using ML and other web techniques. We explained the proposed solution to the best of our abilities.

Ashish Raj
B.Tech. IV Year
Discipline of Mechanical Engineering
IIT Indore

Acknowledgments

We want to thank our B.Tech Project supervisor, Dr. Chandresh Kumar Maurya, for his guidance and constant support in structuring the project and for providing valuable feedback throughout this project. Furthermore for giving us an opportunity to deal with advanced technologies. We would also like to thank all the faculty members of the Discipline of Mechanical Engineering and Discipline of Computer Science and Engineering for their invaluable support during the presentations. We are grateful to the Institute for providing the necessary tools and utilities to complete the project. Lastly, we offer our sincere thanks to everyone who helped us complete this project, whose name we might have forgotten to mention.

Ashish Raj
B.Tech. IV Year
Discipline of Mechanical Engineering
IIT Indore

Abstract

In India, those who reside in rural areas still lack access to basic medical care. The lack of competent radiologists in these places is the primary cause of challenges. Patients are unable to comprehend the main points of radiology reports because of their language and technical complexity, even if the report is provided to them. People must seek out professionals in order to receive proper care, even though they may not even be aware of their true condition.

We intend to employ ML and NLP technologies in our project to address these issues. The patient only needs to upload their x-ray image via a gadget to receive a thorough report in both english and their native tongue. In addition, a summary tool is used to create a brief report that would be beneficial to both patients and clinicians. Additionally, if they have a sickness, we will extract the name of the illness from the report and give them recommendations for doctors to contact further.

Contents

Candidate's Declaration v

Certificate by BTP Guide v

Preface vii

Acknowledgments ix

Abstract xi

List of Figures xv

1 Introduction 1

2 Literature Review 3
- 2.1 Methods of NER . . . 3
  - 2.1.1 Implementations of NER . . . 4
  - 2.1.2 NER for Biomedical terminologies . . . 4
- 2.2 BERT and its limitations in biomedical domain . . . 4
- 2.3 BioBERT . . . 5
- 2.4 BERN2 tool for performing NER . . . 6
- 2.5 Text Summarization using BART . . . 6

3 Description and Implementation Details 7
- 3.1 Optical Character Recognition (OCR) . . . 7
- 3.2 Natural Language Processing (NLP) . . . 8
- 3.3 Named Entity Recognition (NER) . . . 9
- 3.4 What are Transformers? . . . 10
  - 3.4.1 How do Transformers work? . . . 10
- 3.5 BERT . . . 11
  - 3.5.1 What makes it Bidirectional? . . . 13
  - 3.5.2 Masked Language Model (MLM) . . . 13
  - 3.5.3 Next Sentence Prediction (NSP) . . . 14
- 3.6 BioBERT: Pre-Training BERT with biomedical corpus . . . 14
  - 3.6.1 Tokenization in BioBERT . . . 15
  - 3.6.2 Fine-tuning BioBERT . . . 15
- 3.7 BERN2 - A tool built over BioBERT model . . . 16
- 3.8 Medical Report Summarization using BART . . . 18
  - 3.8.1 How actually BART model achieves the summarization task? . . . 18
- 3.9 Recommending Doctors based on Illness . . . 20
  - 3.9.1 Use of BeautifulSoup - a web scraping tool . . . 20

4 Experimental Walkthrough 23
- 4.1 Dataset Description of Fine-Tuning of BioBERT . . . 23
- 4.2 Hardware and Software Specifications . . . 25
- 4.3 Building Applications Backend and the Frontend . . . 25
- 4.4 Important Links . . . 26

5 Output Observations and Results 27

5.1 BioBERT model: Output and Observations 27
5.2 BioBERT model: Performance Analysis 28
5.3 NER Output 28
5.4 BART summarizer: Results 30
5.5 Result of Doctor Recommendations Task 31

6 Conclusion and Future Scope **34**
6.1 Conclusion 34
6.2 Future Scope 34

List of Figures

1 Results showing how BioBERT is different from BERT [6] 5

2 Architecture of Tesseract.js [1] 8

3 Transformer architecture [2] 11

4 BERT Model architecture [3] 12

5 Pre-Training datasets Overview [6] 14

6 Pre-Training datasets Combinations Overview [6] 15

7 Architecture of BERN2 [10] 17

8 Architecture of BART model [7] 19

9 Pre-Training Dataset Overview for BioBERT 24

10 Performance of BioBERT fine-tuned on NCBI-disease(50epochs) 28

11 Performance of BioBERT fine-tuned on BC5CDR-disease(50epochs) 28

12 Raw Output from BERN2 tool 29

13 Final Result for our NER task 30

14 BART summarizer: Working Example (summarizing to English) 30

15 BART summarizer: Working Example (summarizing to Telugu) 31

16 Output of Doctor Recommendations Operation 32

Chapter 1
Introduction

The essential need of our research is the Named Entity Recognition (NER) of sickness from the reports, one of the most common data preparation tasks in Natural Language Processing (NLP). Once the document's keywords containing the disease's name have been retrieved, we may utilise them to propose physicians.

Due to the exponential growth of biomedical papers, biomedical text mining is becoming more important. As Natural Language Processing (NLP) technology progresses, academics are more interested in extracting meaningful information from biomedical literature, and deep learning has sped the development of effective biomedical text mining models. In this study, we employ a disease-specific dataset and the BioBERT model to extract disease names as keywords.

To summarise the medical report, a pre-trained BART model based on denoising autoencoder for pretraining sequence-to-sequence models is used. In order to generate the report in the native language, the summarised report is supplied as input to a translation model. The output of the translation model is beneficial to the patient, since it keeps them informed of the illness (if any) they are suffering from. Google's APIs are used when proposing nearby experts to patients. It is necessary to restructure the output of these APIs so that it is more user-friendly.

In the last section of the study, we share the repository on GitHub that contains the implementation details. We also give a link to a project demonstration video.

Chapter 2

Literature Review

Using a variety of machine learning (ML) and natural language processing (NLP) techniques, we present new efforts on the extraction of bio-medical literature in this chapter. We compare various text summarising methods that are currently in use as part of our literature review. In addition, we will provide the insights of the text summarization models, we read about.

2.1 Methods of NER

Different methods of Named Entity Recognition are discussed as follows:

- One approach involves labelling a lot of data in order to train the model for multi-class classification using various machine learning methods. To deal with the ambiguity of the sentences, the model needs to be labelled in addition to having a thorough grasp of context. This makes it difficult for a straightforward machine learning algorithm to complete.
- Another method is the Conditional Random Field, which is used by both NLTK and NLP Speech Tagger. It's a probabilistic model that works well for modelling sequential data, like words. The sentence's context can be fully understood using the CRF.
- Deep Learning Based NER: Because it can put words together, deep learning NER is substantially more accurate than the prior way. This is because it made use of a technique called word embedding, which can recognise the semantic and syntactic connections between different words. Additionally, it has the capacity to automatically learn and interpret high-level and topic-specific terms. Deep learning NER may now be used to complete a variety of jobs. Researchers, for example, can use their time

more effectively because deep learning can handle most repetitious tasks on its own.

2.1.1 Implementations of NER

Spacy and NLTK are two of the most well-known libraries used to implement named entity recognition.

2.1.2 NER for Biomedical terminologies

Recent improvements in deep learning methods used for natural language processing have enabled the development of biomedical text mining models (NLP). For instance, during the past few years, performance in biological named entity recognition (NER) has significantly improved with the use of Long Short-Term Memory (LSTM) and Conditional Random Field (CRF). The relation extraction (RE) and question answering tasks of biomedical text mining have been improved by other deep learning-based models (QA).

2.2 BERT and its limitations in biomedical domain

However, there are restrictions when using modern NLP approaches for biological text mining. First off, it is challenging to gauge the performance of current word representation models such as **Word2Vec (Mikolov et al., 2013) [4], ELMo (Peters et al., 2018) [9], and BERT (Devlin et al., 2019)** [3] on datasets comprising texts from wide domains (such as Wikipedia). Additionally, there are significant differences in the word distributions between general and biomedical corpora, which can frequently cause issues for biomedical text mining algorithms. Due to this, modern methods for biological text mining heavily rely on modified forms of word representations **(Habibi et al. [5], 2017; Pyysalo et al., 2013 [8])**.

We discover that to be successful in biomedical text mining tasks,

state-of-the-art word representation models, such as BERT, need to be trained on biomedical corpora. Although ELMo and BERT have demonstrated the value of contextualised word representations, they are unable to perform well on biomedical corpora since they have only been trained on broad domain corpora. Numerous biomedical NLP studies could potentially benefit from modifying BERT because it consistently produces excellent results on a variety of NLP tasks while employing nearly the same structure across the tasks.

Task	Dataset	Model	Sample
NER	NCBI disease	BERT	WT1 missense mutations, associated with male pseudohermaphroditism in **Denys–Drash syndrome**, fail to . . .
		BioBERT	WT1 missense mutations, associated with **male pseudohermaphroditism** in **Denys–Drash syndrome**, fail to . . .
	BC5CDR (Drug/Chem.)	BERT	. . . a case of oral **penicillin anaphylaxis** is described, and the terminology . . .
		BioBERT	. . . a case of oral **penicillin** anaphylaxis is described, and the terminology . . .
	BC2GM	BERT	Like the DMA, but unlike all other mammalian class II A genes, the zebrafish gene codes for two cysteine residues . . .
		BioBERT	Like the **DMA**, but unlike all other mammalian class II A genes, the zebrafish gene codes for two cysteine residues . . .
QA	BioASQ 6b-factoid		Q: Which type of urinary incontinence is diagnosed with the Q tip test?
		BERT	A total of 25 women affected by clinical **stress** urinary incontinence (SUI) were enrolled. After undergoing (. . .) Q-tip test, . . .
		BioBERT	A total of 25 women affected by clinical **stress urinary incontinence** (SUI) were enrolled. After undergoing (. . .) Q-tip test, . . .
			Q: Which bacteria causes erythrasma?
		BERT	**Corynebacterium** minutissimum is the bacteria that leads to cutaneous eruptions of erythrasma . . .
		BioBERT	**Corynebacterium minutissimum** is the bacteria that leads to cutaneous eruptions of erythrasma . . .

Note: Predicted named entities for NER and predicted answers for QA are in bold.

Figure 1: Results showing how BioBERT is different from BERT [6]

2.3 BioBERT

Specifically designed for biomedical text mining, BioBERT [6] is a pre-trained language representation model. Applying BERT to the biomedical domain requires that it be pre-trained on biomedical corpora, which is a highly important step. On biomedical text mining tasks including NER, RE, and QA, BioBERT outperforms earlier models while requiring little task-specific architecture adjustment. In several biological text mining tasks, including NER for disease extraction, it has already demonstrated highly good results.

2.4 BERN2 tool for performing NER

The automatic extraction of biological entities (such as illnesses) from the ever expanding body of scientific literature is made possible by named entity recognition (NER) and named entity normalisation (NEN), two crucial tasks in biomedical natural language processing. The BERN2 [10] (Advanced Biomedical Entity Recognition and Normalization) tool uses a multitask NER model and neural network-based NEN models to deliver considerably faster and more accurate inference than the prior neural network-based NER tool. Large-scale biomedical texts can benefit from annotation using this tool for a variety of purposes.

The nine biological entity categories supported by BERN2 are gene/protein, illness, drug/chemical, species, mutation, cell line, cell type, DNA, and RNA. By utilising a single multi-task NER model, the annotation time could be drastically cut.

Before using BERN2 tool, we went through the architectures of other publicly available NER tools like one used in [11], and [13].

2.5 Text Summarization using BART

BART [7] is a transformer encoder-decoder (seq2seq) model that combines an autoregressive (GPT-like) decoder with a bidirectional (BERT-like) encoder. In order to pre-train BART, it first corrupts text using a random noise function and then learns a model to restore the original text. BART uses the same sequence-to-sequence Transformer architecture as BERT, with the exception that we change ReLU activation functions to GeLUs in accordance with GPT. BART has about 10% more parameters overall than the BERT model of the same size. BART is particularly effective when tailored for text generation (such as summarization and translation), but it also performs well for comprehension tasks (e.g. text classification, question answering).

Chapter 3
Description and Implementation Details

This chapter will cover the procedures required in putting the different tools and models that we have developed into practise. For the purpose of turning the medical report image into text, we first use an optical character recognition (OCR) scanning tool. The ML and NLP methods we employed in our project to address the given issue statement are then thoroughly discussed. We will go over each stage, starting with the BERT model's pre-training (5), to fine-tuning it on datasets that may aid us in prediction tasks, applying entity recognition to reports, extracting illness names from tagged labels, and using those in subsequent implementation tasks. We'll provide examples to show how the different algorithms operate. We begin with section 3.1 .

3.1 Optical Character Recognition (OCR)

Optical Character Recognition (OCR) is a technology that optically recognizes handwritten or printed characters as data, and converts them into characters that can be used in computer programs. OCR is used in a variety of fields. For example, OCR is used to detect stolen vehicles by recognizing their license plates, and to digitize printed books.

In our project, we use Tesseract.js to achieve this. Tesseract.js is an open source OCR library that supports over 100 languages. Tesseract.js compiles the Tesseract OCR engine written in C into JavaScript WebAssembly. With Tesseract.js, we can easily build OCR programs that run in the browser. The operation flow of the web application is as follows:

- User uploads an image of medical report.
- Image is OCR-processed with Tesseract.js.

- The recognized text is used in further tasks.

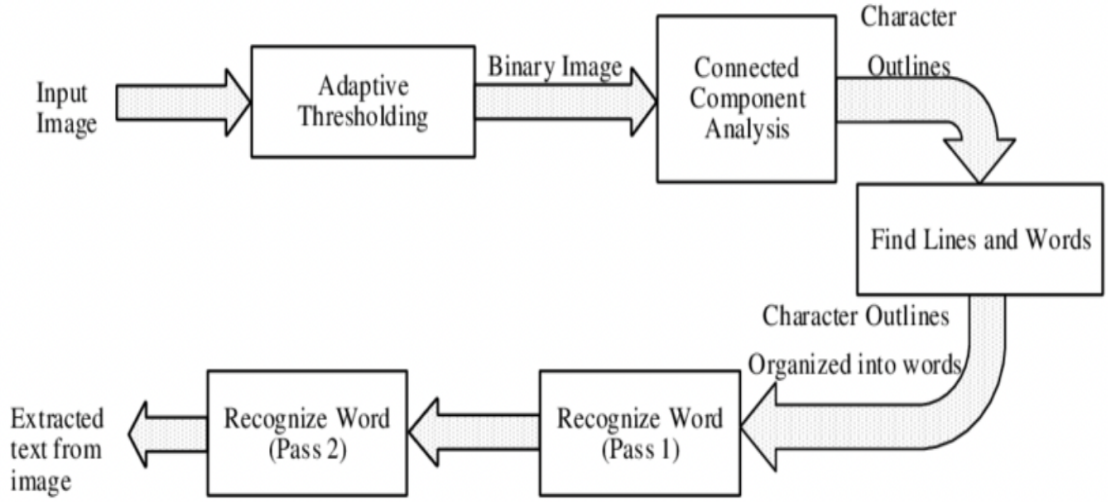

Figure 2: Architecture of Tesseract.js [1]

3.2 Natural Language Processing (NLP)

NLP, which is a branch of AI (AI). In many different commercial disciplines and places, personal assistants employ this technology. The system analyses the verbal input from the user, deconstructs it for accurate comprehension, and then processes it as necessary. Due to the fact that it is a relatively current and successful strategy, there is a huge demand for it right now. It has already been feasible to hold interactive conversations with a human being and work with smart gadgets thanks to advancements in the field of natural language processing. The focus of AI applications in NLP was on knowledge representation, logical reasoning, and constraint fulfilment. Here,

it was used first for semantics and then for grammar. The field is divided into three different parts:

- Speech Recognition: The translation of spoken language into text.
- Natural Language Understanding (NLU): The computer's ability to understand what we say.
- Natural Language Generation (NLG):The generation of natural language by a computer.

3.3 Named Entity Recognition (NER)

A sub-task of information extraction called named-entity recognition (NER) aims to find and categorise named entities in text into pre-defined categories, such as the names of people, organisations, locations, expressions of times, quantities, monetary values, percentages, etc.

Both statistical approaches like machine learning and linguistic grammar-based strategies have been used to construct NER systems. Better accuracy is often achieved by hand-crafted grammar-based systems, albeit at the expense of recall and months of labour by expert computational linguists. A significant amount of manually annotated training data is often needed for statistical NER systems. Semi-supervised methods have been proposed as a way to reduce some of the annotation work. The two steps that make up the fundamental process of NER are as follows:

- Identifying items in the text.
- Putting them into various categories.

The model is then presented with the unlabeled text and asked to predict. We can provide the model with feedback on its prediction in the form of an error gradient of the loss function, which measures the discrepancy between the training example and the anticipated result, because we are aware of the

right response. The gradient and model updates are increasingly significant the larger the difference.

3.4 What are Transformers?

ML training can be efficiently parallelized thanks to the Transformer design [12]. This makes it possible to train BERT on vast volumes of data in a short amount of time because to massive parallelization.

Transformers employ an attention mechanism to notice how words relate to one another. Transformers are used in NLP models all around the world because to an idea that was first put out in the well-known Attention Is All You Need paper from 2017.

3.4.1 How do Transformers work?

The Transformer architecture [12] allows for effective parallelization of ML training. Due to extensive parallelization, this allows BERT to be trained on large amounts of data quickly. Transformers use an attention mechanism to take note of the connections between words. As a result of a notion that was initially presented, transformers are employed in NLP models all around the world.

Attention, a potent deep-learning method initially used in computer vision models, is the key to how transformers operate. Similar to the way people digest information by paying attention. We are remarkably adept at forgetting or disregarding innocuous daily stimuli that don't threaten us or call for a reaction on our part.

Similar to this, machine learning models must learn to focus solely on the important information and avoid wasting time processing unimportant data. Transformers provide differential weights that indicate which words in a phrase need to be processed more quickly.

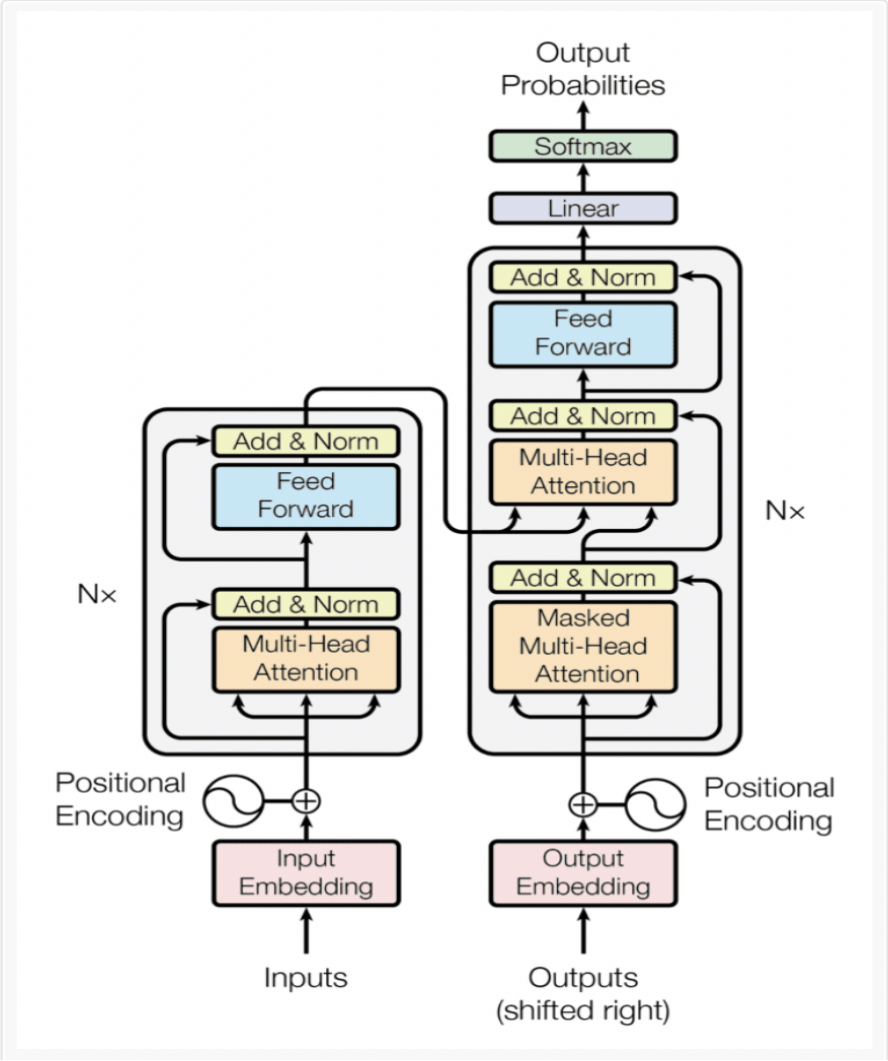

Figure 3: Transformer architecture [2]

3.5 BERT

Google's language representation model is called BERT [3], which stands for Bidirectional Encoder Representations from Transformers. It employs two procedures—pre-training and fine-tuning—to build cutting-edge models for a

variety of activities.

Its unique characteristic is the unified design across many downstream activities, which we will examine in more detail later. Accordingly, the same pre-trained model can be adjusted for a range of final tasks that might not be identical to the task the model was trained on and still produce results that are close to cutting-edge. We first simultaneously train the model on the pre-training tasks. The same model can be adjusted for a number of downstream jobs once the pre-training is finished. It should be noted that a different model is tailored for a certain downstream activity. Therefore, after fine adjustment, a single pre-trained model can produce several downstream task-specific models.

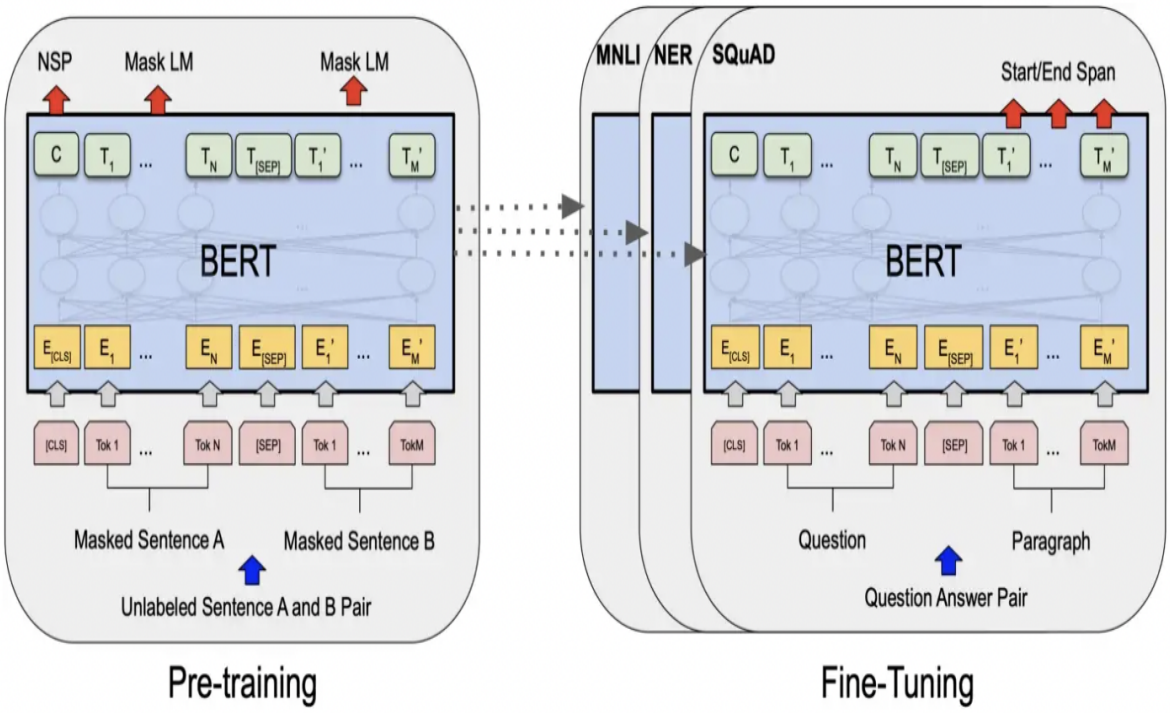

Figure 4: BERT Model architecture [3]

3.5.1 What makes it Bidirectional?

Typically, we train a language model on unrelated tasks that aid in the development of a contextual knowledge of the words in the model. Predicting the next word or words that will be said next is a common challenge in this category. Since each word would be able to indirectly "see itself," you would already be somewhat aware of what to expect when you approached the same sentence from the opposite way, such training methods cannot be expanded and utilised for bidirectional models. a data breach instance.

In this case, the model could guess the intended term with ease. Furthermore, we can't ensure that the model, even after being fully trained, has taken into account the context of the words and isn't merely concentrating on improving the obvious predictions. So how does BERT accomplish bidirectional pre-training? The method it use to do this is Masked Language Modelling.

3.5.2 Masked Language Model (MLM)

BERT stands for Bidirectional Encoder Representations from The deep bidirectional learning capability of the model is enabled by this job. In this job, a certain proportion of the input tokens are randomly mask (Replaced with [MASK] token) and the model only tries to predict these mask tokens, not the complete input sequence. The final output words are obtained by feeding the model's predicted tokens into an output softmax over the vocabulary.Due to the fact that most downstream tasks do not include predicting masked words, this causes a mismatch between the pre-training and fine-tuning tasks. A slight change in the way we disguise the input tokens helps to counteract this.

3.5.3 Next Sentence Prediction (NSP)

The link between two sentences, which is important in many downstream tasks like Question Answering (QA) and Natural Language Inference (NLI), is not explicitly captured by the Language Modelling.

3.6 BioBERT: Pre-Training BERT with biomedical corpus

First, we pre-train BioBERT [6] using weights from BERT, which were then used to initialise BioBERT. Then, BioBERT is prepared using corpora from the biomedical area (PubMed abstracts and PMC full-text articles). BioBERT is improved and tested on three common biomedical text mining jobs to demonstrate the effectiveness of our technique in this field (NER, RE and QA).

Corpus	Number of words	Domain
English Wikipedia	2.5B	General
BooksCorpus	0.8B	General
PubMed Abstracts	4.5B	Biomedical
PMC Full-text articles	13.5B	Biomedical

Figure 5: Pre-Training datasets Overview [6]

The model BioBERT achieves state-of-the-art performance on a variety of biomedical text mining tasks while requiring only minor architectural alterations, in contrast to most prior biomedical text mining models that are primarily focused on a specific job, such as NER or QA.

BERT was pre-trained on the English Wikipedia and BooksCorpus to

serve as a general-purpose language representation model. However, a lot of terms and proper nouns that are only understood by biomedical researchers can be found in texts in the biomedical domain. As a result, NLP models created for general-purpose language comprehension frequently perform poorly in tasks involving biomedical text mining.

Model	Corpus combination
BERT (Devlin *et al.*, 2019)	Wiki + Books
BioBERT (+PubMed)	Wiki + Books + PubMed
BioBERT (+PMC)	Wiki + Books + PMC
BioBERT (+PubMed + PMC)	Wiki + Books + PubMed + PMC

Figure 6: Pre-Training datasets Combinations Overview [6]

3.6.1 Tokenization in BioBERT

For tokenization, BioBERT uses WordPiece tokenization , which mitigates the out-of-vocabulary issue. With WordPiece tokenization, any new words can be represented by frequent subwords. The following embeddings are added to token embeddings:

- Segment Embedding: They provide information about the sentence a particular token is a part of.
- Position Embedding: They provide information about the order of words in the input.

3.6.2 Fine-tuning BioBERT

By switching out the right inputs or outputs, many downstream processes

may be fine-tuned. Typically, to train task-specific models, we extend the BERT with an additional output layer and then fine-tune the resulting model's whole set of parameters. A benefit of adding input/output layers while leaving the BERT model alone is that fewer parameters must be learnt from beginning, making the process quick, inexpensive, and resource-effective.

BioBERT is a model that can be used for a variety of downstream text mining tasks with only minor architectural modification. BERT features a straightforward design based on bidirectional transformers, unlike the majority of earlier studies, which were constructed using various combinations of LSTMs and CRFs. BERT computes just token level BIO2 probability using a single output layer based on representations from its final layer.

3.7 BERN2 - A tool built over BioBERT model

The precise locations and kinds of biological named entities in provided plain text are initially extracted using a multi-task NER model of BERN2 [10]. The NER model, for instance, locates the places of the words "tumour" and "arginine" in the text and classifies them as diseases and drugs/chemicals, respectively, for the plain text "tumour development through arginine." To enhance the number of entities that are successfully normalised, we apply hybrid NEN models for three entity kinds (gene/protein, illness, and drug/chemical) . The annotations of PubMed articles (abstracts only) that are kept in an external database are pre-computed using the plain text annotation method that was previously disclosed.

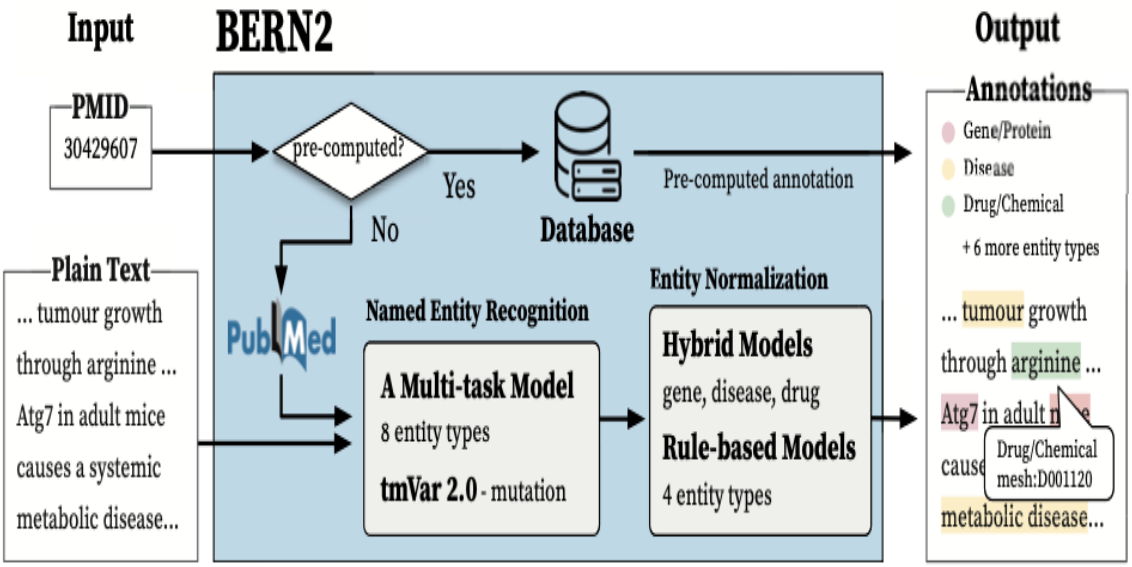

Figure 7: Architecture of BERN2 [10]

As a result, BERN2 delivers pre-computed annotations from its database whenever it is feasible when PMIDs are provided. This is substantially faster than annotating the bare text, as seen in Table 1. BERN2 annotates a PubMed article's abstract if one hasn't already been pre-computed (i.e., it hasn't been located in the database) and puts the annotation results in the database. The multi-task NER model used by BERN2 offers eight entity classes' efficient parallel inference (except for mutation). Our NER model accepts a text as an input and generates predictions from all of the task-specific layers concurrently during the inference stage. In actuality, the multi-task NER model in BERN2 uses very little GPU RAM.

According to the research report used to develop the BERN2 tool, we have implemented it. The implementation causes it to run in a local environment that can be accessed using RESTful api. In the unlikely event that the system you are working on is incompatible, BERN2 is also available as a free open api.

3.8 Medical Report Summarization using BART

Sequence-to-sequence model BART [7] was developed as a denoising autoencoder. This kind of approach is applicable for text summarising (providing a summary of or paraphrasing a lengthy text document), question-answering (generating answers for a given question on a certain corpus), and sequence classification (categorizing input text sentences or tokens). In tasks, such as text summarization, that require natural language understanding (or NLU) it is imperative for our model to read the text as a whole and understand each token in the context of what came, both, before and after it.

We selected HuggingFace Transformer models as our summarizer tool since they offer a simple-to-use implementation of some of the best-performing models in natural language processing for tasks like text generation and text summarization. We have fairly satisfactory results from our physical examination for the summary of the medical report.

3.8.1 How actually BART model achieves the summarization task?

In order to determine the optimum representation of its input sequence, the first component of BART uses the bi-directional encoder of BERT. The BERT encoder outputs a vector comprising sentence-level information in addition to an embedding vector for each token in each text sequence in its input. By learning for both token- and sentence-level tasks in this way, the decoder becomes a solid starting point for any upcoming fine-tuning tasks.

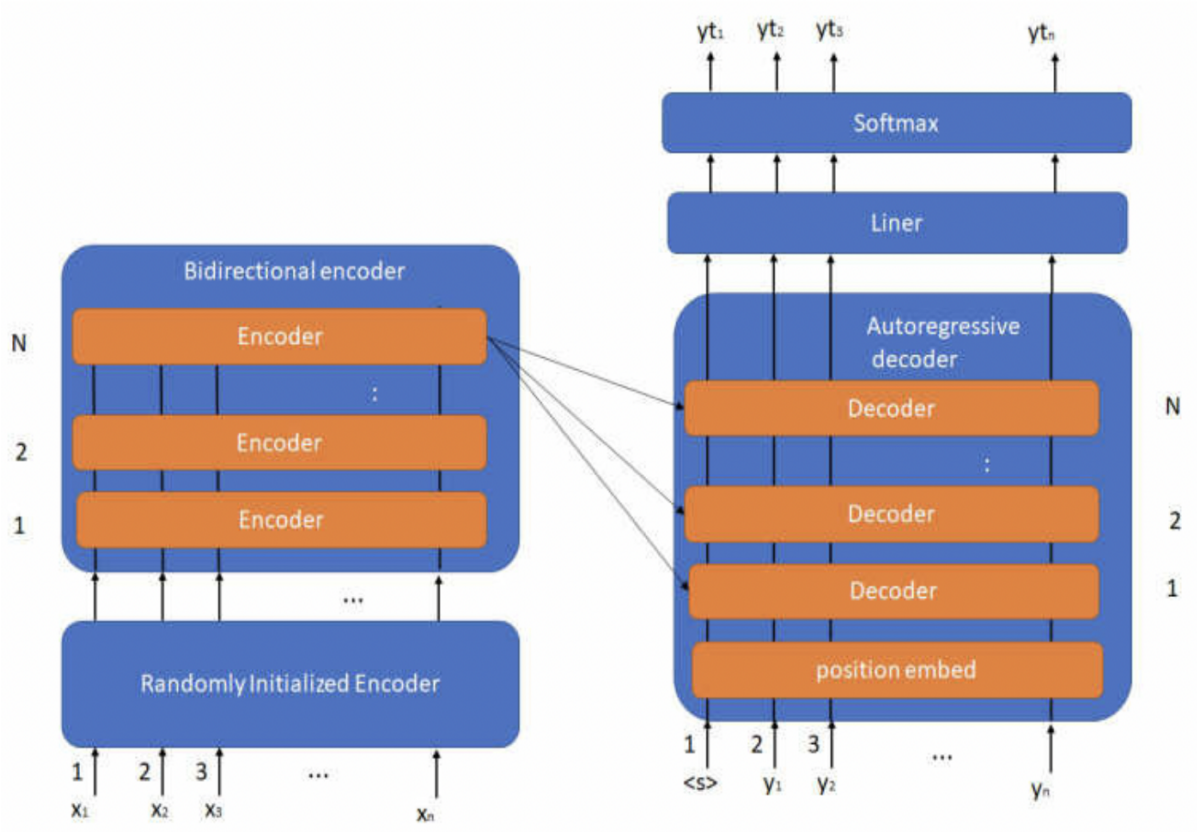

Figure 8: Architecture of BART model [7]

A decoder must then comprehend these in order to map them to the output target. However, given that the model depends on a more thorough input prompt, activities like token prediction or next sentence prediction could perform badly when utilising a decoder of a similar architecture. In these scenarios, model architectures that can only look at the previous words in the sequence during training are necessary. Consequently, a causal or autoregressive model that uses just historical data to make predictions about the future is useful.

The input to the decoder would still be a learnt representation (or embedding) of the original text sequence even though the encoder and decoder have been separated. In order to establish a denoising auto-encoder

architecture, BART connects the bi-directional encoder to the autoregressive decoder.

3.9 Recommending Doctors based on Illness

Assuming we know where they are located, we will use the disease names we collected from the medical report to recommend doctors to patients. In order to match patients with nearby doctors, we need location as a criteria (like doctors in same city or same locality). We use two apis and a few web scraping techniques to do this:

- Google Search Api - The Custom Search JSON API lets us develop websites and applications to retrieve and display search results from Programmable Search Engine programmatically. With this API, we can use RESTful requests to get web search results in JSON format.
- Google Maps Api - The functions in the Places Library, Maps JavaScript API allow your application to look for places (defined in this API as establishments, geographic locations, or notable points of interest) located within or around a fixed point or within the boundaries of a defined area, such as the map's boundaries.

3.9.1 Use of BeautifulSoup - a web scraping tool

Python's BeautifulSoup library can extract data from HTML and XML files. It provides natural means of traversing, searching, and altering the parse tree in conjunction with your preferred parser. Programmers frequently end up saving hours or even days of effort. Data scraping is a technique used to gather information from websites. It is also known as online harvesting or web data extraction. It's a type of copying where specific information is acquired and copied from the internet, usually into a central local database or spreadsheet, for subsequent retrieval or analysis.

A web page is retrieved and extracted from in order to scrape it. To gather pages for later processing, web crawling is a key part of web scraping. After being fetched, extraction can happen. The data on a page can be copied into a spreadsheet or imported into a database after it has been processed, searched, and reformatted. Web scrapers often steal something from a page so they can use it elsewhere for another reason.

For our project, we use BeautifulSoup because of the documentation's accessibility and simplicity of usage.

Chapter 4
Experimental Walkthrough

In this chapter, we will explain the datasets used and the implementations setup, along with the specifications of our system used. The section 4.1 will mention the various datasets used to train and test our proposed method along with a brief overview of its contents.

Our project's fundamental concept is built on the tasks of text summarization and named entity recognition (NER). In order to extract the name of the sickness or ailment from the medical report that we are given, our implementation of NER is built on BERN2, a tool that leverages the BioBERT model on the inside. Because BERN2 is based on BioBERT, we describe the datasets we used to fine-tune BioBERT in this chapter. In addition, the very first stage of our project uses optical character recognition (OCR) to turn the photos of the medical reports into text that will be used in subsequent processes. To implement OCR in our software, we make use of the potent Tesseract.js package. We will also discuss Tesseract.js' implementation in detail. In addition, we will give implementation information for the web scraping strategies we have employed, in this chapter.

4.1 Dataset Description of Fine-Tuning of BioBERT

With the help of three disease-extraction datasets, we train and test BioBERT in order to evaluate its performance for NER-based biomedical text recognition. Even though BioBERT is the BERT model that was built by pre-training it using a combination of the Pubmed and PMC datasets, we have enhanced it to make it even more effective at the task of disease extraction.

Corpus	Number of words	Domain
English Wikipedia	2.5B	General
BooksCorpus	0.8B	General
PubMed Abstracts	4.5B	Biomedical
PMC Full-text articles	13.5B	Biomedical

Figure 9: Pre-Training Dataset Overview for BioBERT

The two datasets used for fine tuning BioBERT are:

- NCBI-disease
- BC5CDR-disease

NCBI Disease: A collection of 793 PubMed abstracts that have been thoroughly annotated at the mention and idea levels for the benefit of the community of biomedical natural language processors. Each PubMed abstract was manually annotated with references to diseases by two annotators. The 6,892 illness mentions in the public version of the NCBI disease corpus are linked to 790 different disease categories. A training, testing, and development set of the corpus has been produced.

An essential component in allowing the creation of complicated information retrieval tasks that link diseases to treatments, causes, or other forms of information is the annotation of every sentence in a PubMed page (title and abstract) that is included in the NCBI disease corpus.

BC5CDR-disease: There are 4,409 annotated chemicals, 5,818 disorders, and 3,116 chemical-disease interactions in these 1,500 PubMed entries. The mention text spans and normalised concept Identifiers are both included in each annotation. The entities were separately recorded by two annotators to guarantee correctness.

4.2 Hardware and Software Specifications

All the algorithms and models were coded in python-3.7, and written on local machine using VS Code IDE. We also took help of Google Colab to test individual component of our project independently. The BioBERT model was developed using TensorFlow-2.8, Keras-2.8, and PyTorch-1.10. While using model, a remote machine containing Cuda-11.4 toolkit and Tensorflow-GPU-1.14 toolkit was used to enhance the speed of computation.

4.3 Building Applications Backend and the Frontend

For the backend of our web application, we built a server environment using Node.js and Python (Flask). By opening an end point in the Node.js server, which serves requests made by the frontend with simply a mouse click, the disease extraction portion of the project is launched. We utilised the Flask server to provide the text-summarization and doctor suggestions jobs since it was simple to construct (Web scraping part). We used HTML and CSS to construct the frontend because it has a simple design.

4.4 Important Links

- **GitHub Repository Link**
- **Demo Video Link**

Chapter 5

Output Observations and Results

In this chapter, we'll look at the results of the implementations we used in our project. We shall discuss the BioBERT model's correctness. Along with its translation into a local language, we will also observe the outcomes of the text-summarization strategies we used. Additionally, we will record the results of the instruments we utilised to recommend doctors. We start with section 5.1

5.1 BioBERT model: Output and Observations

Below, we present the difference in the output we get from the BERT and BioBERT model, which signifies the importance of using BioBERT over BERT for the biomedical domain.

In the below examples, the bold and italic words are the one which tagged as a biomedical term (in terms of BIO tags), as part of the Named Entity Recognition (NER) task. Also, the first text is from BERT and the second one is from BioBERT model.

1. Fine-tuned on NCBI disease dataset :
   - WT1 missense mutations, associated with male pseudohermaphroditism in ***Denys–Drash syndrome***, fail to ...
   - WT1 missense mutations, associated with ***male pseudohermaphroditism*** in ***Denys–Drash syndrome***, fail to ...
2. Fine-tuned on BBC5CDR disease dataset:
   - Like the DMA, but unlike all other mammalian class II A genes, the zebrafish gene codes for two cysteine residues.
   - Like the ***DMA***, but unlike all other mammalian class II A genes, the zebrafish gene codes for two cysteine residues.

We can see that the BioBERT's output is more accurately labelled than BERT's. We can observe that BERT overlooked terms like ***"pseudohermaphroditism"*** in the first case and ***"DMA"*** in the second case, on analyzing only one line of text as an example. These two terms are crucial in the biomedical terminology. The BioBERT model correctly identified every medical phrase in the aforementioned literature.

5.2 BioBERT model: Performance Analysis

Below the two images gives us the performance of the BioBERT model on running on NCBI-disease, and BC5CDR-disease dataset individually:

```
(ashish_env) mohit@aiml:~/ashish2/ashish$ perl biocodes/conlleval.pl < $OUTPUT_DIR/NER_result_conll.txt
processed 24520 tokens with 962 phrases; found: 7230 phrases; correct: 121.
accuracy:  63.54%; precision:   1.67%; recall:  12.58%; FB1:   2.95
             MISC: precision:   1.67%; recall:  12.58%; FB1:   2.95  7230
```

Figure 10: Performance of BioBERT fine-tuned on NCBI-disease(50epochs)

```
(ashish_env) mohit@aiml:~/ashish2/ashish$ perl biocodes/conlleval.pl < $OUTPUT_DIR/NER_result_conll.txt
processed 124750 tokens with 4424 phrases; found: 28639 phrases; correct: 550.
accuracy:  46.51%; precision:   1.92%; recall:  12.43%; FB1:   3.33
             MISC: precision:   1.92%; recall:  12.43%; FB1:   3.33  28639
```

Figure 11: Performance of BioBERT fine-tuned on BC5CDR-disease(50epochs)

5.3 NER Output

Our main task of the Named Entity Recognition (NER) part, was to identify disease name from the medical report. The figure 12 provides us with the raw output we get on applying BioBERT model, utilising BERN2 as an intermediary tool.

```
Success:                                                                    results.js:17
▼{annotations: Array(5), text: 'FINAL REPORT ic PORTABLE AP CHEST FILM __ AT ___ C… pneumonia. No pl
eural effusions or pneumothorax.', timestamp: 'Sat Nov 26 20:15:15 +0000 2022'} i
  ▼annotations: Array(5)
    ▼0:
      ▶id: ['mesh:D001261']
       is_neural_normalized: false
       mention: "atelectasis"
       obj: "disease"
       prob: 0.5966000556945801
      ▶span: {begin: 692, end: 703}
      ▶[[Prototype]]: Object
    ▶1: {id: Array(1), is_neural_normalized: true, mention: 'airspace consolidation', obj: 'disease'
    ▶2: {id: Array(1), is_neural_normalized: false, mention: 'pneumonia', obj: 'disease', prob: 0.99
    ▶3: {id: Array(1), is_neural_normalized: false, mention: 'pleural effusions', obj: 'disease', pr
    ▶4: {id: Array(1), is_neural_normalized: false, mention: 'pneumothorax', obj: 'disease', prob: 0
     length: 5
    ▶[[Prototype]]: Array(0)
    text: "FINAL REPORT ic PORTABLE AP CHEST FILM __ AT ___ CLINICAL INDICATION: __ -year-old with na
    timestamp: "Sat Nov 26 20:15:15 +0000 2022"
  ▶[[Prototype]]: Object
```


Figure 12: Raw Output from BERN2 tool

The output we see in figure 12 needs to be destructured to identify illness terms. To convert this raw output to a meaningful form, we used javascript tools. The figure 13 shows us how the new outputs looks like.

The output from BERN2 may well be generalised into a group by doctors, who are under a lot of work pressure, with only one glance. Doctors won't need to read through a detailed report in its entirety as much thanks to this.

After this, the disease names extracted can be used to recommend doctors, in our further steps.

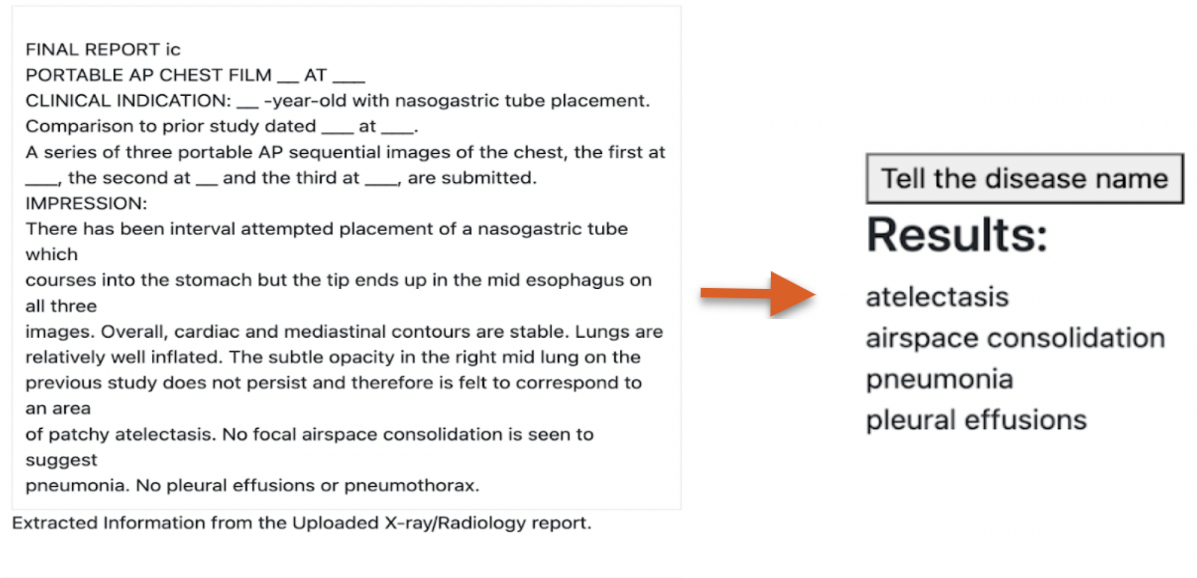

Figure 13: Final Result for our NER task

5.4 BART summarizer: Results

Figure 14 shows us the output of the summarization task, we have done using the Huggingface BART summarizer. The summarization is

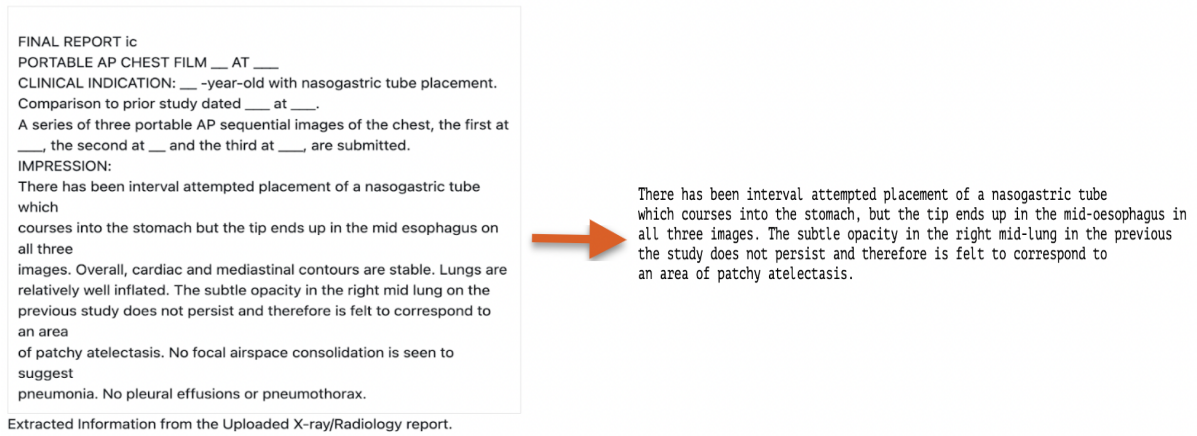

Figure 14: BART summarizer: Working Example (summarizing to English)

Figure 15 shows us the output of the summarization task, which summarizes the medical report text in a local language (here, Telugu). We do it by feeding the summarized report in english (present as output of the BART summarizer in above step) and feeding it into the translator model (implemented by my colleague).

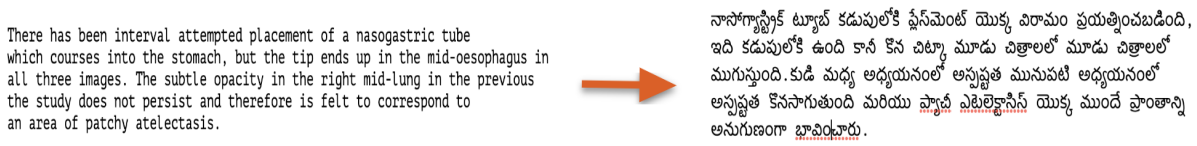

Figure 15: BART summarizer: Working Example (summarizing to Telugu)

5.5 Result of Doctor Recommendations Task

Please find below Figure 16 image of the output we get in the doctor recommendation solution.

By web scraping the Google-Search API results, we arrive at these recommendations. In order to take use of the user's location, we have additionally included the Google Map API. To use this functionality, we must supply a valid Google API key. If the Google API key is not supplied, the application will continue to work, but then, the recommendations will be irrespective to the user's location.

FINAL REPORT ic
PORTABLE AP CHEST FILM __ AT ___
CLINICAL INDICATION: __ -year-old with nasogastric tube placement.
Comparison to prior study dated ___ at ___.
A series of three portable AP sequential images of the chest, the first at ___, the second at __ and the third at ___, are submitted.
IMPRESSION:
There has been interval attempted placement of a nasogastric tube which
courses into the stomach but the tip ends up in the mid esophagus on all three
images. Overall, cardiac and mediastinal contours are stable. Lungs are relatively well inflated. The subtle opacity in the right mid lung on the previous study does not persist and therefore is felt to correspond to an area
of patchy atelectasis. No focal airspace consolidation is seen to suggest
pneumonia. No pleural effusions or pneumothorax.

Extracted Information from the Uploaded X-ray/Radiology report.

Recommend Doctor

Doctor's recommendations:

- Dr. Shailesh Agrawal
  Apollo Hospitals, Vijay Nagar,Indore
- Dr. Pramod Jhawar
  Jhawar Chest, Allergy, Asthma & Bronchoscopy, Sleep Study center, Old Palasia,Indore
- Dr. Salil Bhargava
  Gyanpushp Research Center For Chest And Allergy Diseases, Indore City-2,Indore
- Dr. Gaurav Gupta
  Choithram Hospital & Research Centre, Manikbagh,Indore
- Dr. Nagin Jain
  All Cure Allergy & Chest Centre, Sapna Sangeeta,Indore
- Dr. Ravi Ashok Dosi
  SAIMS AND MOHAK HOSPITAL, Bhanwrasala,Indore
- Dr. Suraj Verma
  CHL Hospital, LIG Colony,Indore
- Dr. Tapan Surana
  Curewell Hospital, New Palasia,Indore

Figure 16: Output of Doctor Recommendations Operation

Chapter 6

Conclusion and Future Scope

In the current chapter, we will discuss the conclusion derived from the results obtained. We will also discuss the future scope of the project.

6.1 Conclusion

The lack of competent radiologists in the remote area is a primary concern in providing rapid treatment to the patients. Patients must seek out professionals in order to receive proper care, even though they may not even be aware of their true condition.

In order to address these issues, we plan to use ML and NLP technologies in this project. To obtain a comprehensive report in both English and their local tongue, the patient merely has to submit their x-ray image using a device. Additionally, a quick report that would be helpful to patients and physicians is produced using a summary tool. Additionally, if they are ill, we will identify it from the report and provide them with referrals for doctors to speak with further.

6.2 Future Scope

The integration of the Image to Text Conversion model will receive the majority of attention in the future (X-ray image to Radiology report generation, which is also based on BERT architecture). This will also entail improving the application's UI and assembling extra server endpoints to handle a variety of different user requests.

Even though we carried out the process with the utmost seriousness, we saw that the accuracy and recall values during the pre-training of the BioBERT

model fell short of what we had anticipated. However, the effects that we can really see are pretty encouraging. The BioBERT model's assessment matrices require improvement. Speech to Speech (Translation from one language to another) is currently under development and is anticipated to be finished shortly.

References

[1] MS Windows NT kernel description. https://www.researchgate.net/figure/Architechture-of-Tesseract-OCR_fig3_328997563. Accessed: 2010-09-30.

[2] MS Windows NT kernel description. https://medium.com/inside-machine-learning/what-is-a-transformer-d07dd1fbec04. Accessed: 2010-09-30.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[4] Yoav Goldberg and Omer Levy. word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*, 2014.

[5] Maryam Habibi, Leon Weber, Mariana Neves, David Luis Wiegandt, and Ulf Leser. Deep learning with word embeddings improves biomedical named entity recognition. *Bioinformatics*, 33(14):i37–i48, 2017.

[6] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240, 2020.

[7] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *CoRR*, abs/1910.13461, 2019.

[8] SPFGH Moen and Tapio Salakoski2 Sophia Ananiadou. Distributional semantics resources for biomedical text processing. *Proceedings of LBM*, pages 39–44, 2013.

[9] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[10] Mujeen Sung, Minbyul Jeong, Yonghwa Choi, Donghyeon Kim, Jinhyuk Lee, and Jaewoo Kang. Bern2: an advanced neural biomedical named entity recognition and normalization tool. *arXiv preprint arXiv:2201.02080*, 2022.

[11] Betty van Aken, Jens-Michalis Papaioannou, Manuel Mayrdorfer, Klemens Budde, Felix A Gers, and Alexander Löser. Clinical outcome prediction from admission notes using self-supervised knowledge integration. *arXiv preprint arXiv:2102.04110*, 2021.

[12] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[13] Xuan Wang, Yu Zhang, Xiang Ren, Yuhao Zhang, Marinka Zitnik, Jingbo Shang, Curtis Langlotz, and Jiawei Han. Cross-type biomedical named entity recognition with deep multi-task learning. *Bioinformatics*, 35(10):1745–1752, 2019.