Medical Textual Question Answering System

PROJECT-II (CS57004) report submitted to

Indian Institute of Technology Kharagpur

in partial fulfilment for the award of the degree of

Master of Technology

in

Computer Science and Engineering

by

Ashish Gour

(18CS30008)

Under the supervision of

Prof. Pawan Goyal

Department of Computer Science and Engineering

Indian Institute of Technology Kharagpur

Spring Semester, 2022-23

May, 2023

DECLARATION

I certify that

(a) The work contained in this report has been done by me under the guidance of my supervisor.

(b) The work has not been submitted to any other Institute for any degree or diploma.

(c) I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.

(d) Whenever I have used materials (data, theoretical analysis, figures, and text) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references. Further, I have taken permission from the copyright owners of the sources, whenever necessary.

Date: May, 2023 (Ashish Gour)

Place: Kharagpur (18CS30008)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR

KHARAGPUR - 721302, INDIA

CERTIFICATE

This is to certify that the project report entitled "**Medical Textual Question Answering System**" submitted by **Ashish Gour** (Roll No. 18CS30008) to Indian Institute of Technology Kharagpur towards partial fulfilment of requirements for the award of degree of Master of Technology in Computer Science and Engineering is a record of bona fide work carried out by him under my supervision and guidance during Spring Semester, 2022-23.

Date: May, 2023

Place: Kharagpur

Prof. Pawan Goyal
Department of Computer Science and Engineering
Indian Institute of Technology Kharagpur
Kharagpur - 721302, India

Abstract

Name of the student: **Ashish Gour** Roll No: **18CS30008**

Degree for which submitted: **Master of Technology**

Department: **Department of Computer Science and Engineering**

Thesis title: **Medical Textual Question Answering System**

Thesis supervisor: **Prof. Pawan Goyal**

Month and year of thesis submission: **May, 2023**

With the outbreak of COVID-19, more and more people are purchasing drugs from online pharmacy stores and seeking online health consulting, asking about their queries about drugs and health problems on such platforms. TATA's 1mg is one such online pharmacy store. Solving these queries on time is essential. The credibility of answers also plays a major role in this demand as sometimes the answers can save the lives of people. In response to this demand, the medical question-answering task has become an important part of natural language processing. In this project, we dealt with the problem by first classifying queries in the context of the diseases and symptoms and then answering them. We started with the five most common diseases/symptoms that patients face, giving us six classes for the classification, including one "other" class for all diseases we didn't consider. We fine-tuned our model and were able to achieve a classification accuracy of 95% with a macro F1 score of 0.87 and able to retrieve significant answers to newly asked patient queries.

Acknowledgements

I would like to thank Prof. Pawan Goyal and Mr. Ankan Mullick for their constant support and help at every step of my Project. Furthermore, I would like to thank the Department of Computer Science and Engineering for the facilities and support provided, which were essential for the Project.

Contents

List of Figures

List of Tables

Abbreviations

MQA	**M**edical **Q**uestion **A**nswer
QAS	**Q**uestion **A**nswering **S**ystem
NLP	**N**atural **L**anguage **P**rocessing
NER	**N**amed **E**ntity **R**ecognization
NLI	**N**atural **L**anguage **I**nference
EHR	**E**lectronic **H**ealth **R**ecords

Chapter 1

Introduction

1.1 Background and Motivation

In recent years, there has been an uptick in the popularity of online platforms. People are using the internet for doing activities from shopping to learning new things. The lockdowns due to the COVID pandemic have also come as a boon for all e-platforms. Online pharmacy is also becoming more popular day by day. There are many pharmaceutical e-platforms as well that not only provide medicines but also online consultation to patients like TATA 1mg etc, where people can ask their queries and get answered by specialized doctors. However, consulting a doctor online can take a long time, and solving patient queries on time is critical for several reasons: Patients who receive prompt and credible solutions to their queries are more likely to have better health outcomes. Delayed or inadequate responses to patient queries can lead to increased stress and anxiety and even worsening of their condition. By getting early and accurate consultations, patients and healthcare providers can reduce the chances of re-admissions, emergency room visits, and unnecessary medical procedures.

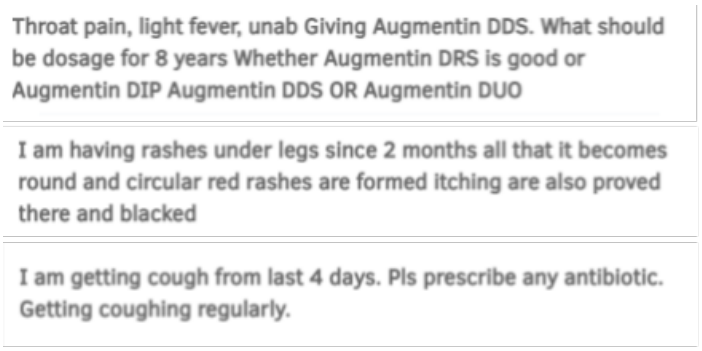

FIGURE 1.1: Examples of the medical query. These queries are asked by patients on TATA's 1 mg website

1.2 Problem Statement and Solution Overview

In cities, doctors be available 24/7 for emergency consulting, but all the activities in a hospital are done manually. The patients need to register then the doctor can view the patient details and provide the required treatment based on the symptoms. All this takes time. And, in rural areas, finding a doctor in an emergency is next to impossible. People need to travel miles for a basic consultation. It takes a lot of human effort and time.

Despite the abundant medical information available online, finding credible and reliable answers to health-related queries remains a significant challenge. The proposed solution addresses the aforementioned queries and issues. It processes the user queries and returns the relevant answers. The credibility of an answer to a Medical query is most when it is answered by a specialist doctor. However, Natural Language Processing has emerged as a promising approach to help individuals navigate and understand medical information. Many times patients have similar queries or queries to previously asked ones. If doctors are not available or reachable at the required time, the patient has to wait for their availability. To address this problem and reduce the wait time, NLP models can be trained to provide fast and accurate solutions. The model can be trained on past data on queries and their solutions.

The effectiveness of NLP in answering medical queries depends on the quality of the underlying dataset, the algorithm used, and the ability to handle complex medical terminologies. Considering the criticality, answering medical queries incorrectly can sometimes result in life-threatening situations and therefore the risk should be minimized.

In this project, we aim to:
1. Improving the accuracy of the classification model to accurately classify the disease class of medical queries
2. Retrieval of the top k significant and credible answers
3. Detection of entities into two categories: drugs and diseases/symptoms to get insights into drug usage and the types of diseases and disorders

For getting the best results, the system first needs to map the query to the most relevant disease class and the class's relevant categories. Further, it needs to search for the significant answers provided by the doctor previously available in those categories. For example, a patient's query is "Allergic severe cough in every winter". This query is classified into the "cold and cough" class and into "cough", "throat pain", and "cold" categories. Based on already answered queries of these categories, it will retrieve the most relevant k answers. Figure 1.2 shows the top k=10 retrieved answers for the given query. Highlighted answers (Answer 2, Answer 3, and Answer 4) are the top 3 annotated ranked answers.

Query	Allergic severe cough in every winter
Answer 1	I usually do allergy testing for this.And give immunotherapy which give permanent solution
Answer 2	Allergic cough can be managed with antiallergy medicines and increasing immunity. Start treatment
Answer 3	Allegra M twice daily for 7 days
Answer 4	It appears as allergic cause. get one srum Ig E levels and coninue allegra
Answer 5	syrup Ascoril LS 5ml 3 times for 5 days
Answer 6	Take tab Dolo sos Tab pantop before breakfast Tab montair LC at nightGet CBC ESR IgE levels tested and chest examined by a md doctor
Answer 7	It is mostly allergic, Put some cow ghee in your nostrils and start some exercise daily, start Tab. Histantin(Kerala Ayurveda) 1-0-1
Answer 8	Get regular cleaning of AC filters. Avoid direct exposure to AC air
Answer 9	you are having allergic rhinitis. These remain under control till you take medication. Permanent solution can be by allergy skin prick testing to find out the causative factor and immunotherapy can be given accordingly
Answer 10	Alex cough syp two tsf daily

FIGURE 1.2: Retrieved k=10 answers for query "Allergic severe cough in every winter" (highlighted answers are top 3 annotated ranked answers)

Figure 1.3 illustrates the general architecture of retrieving top k answers for the query asked.

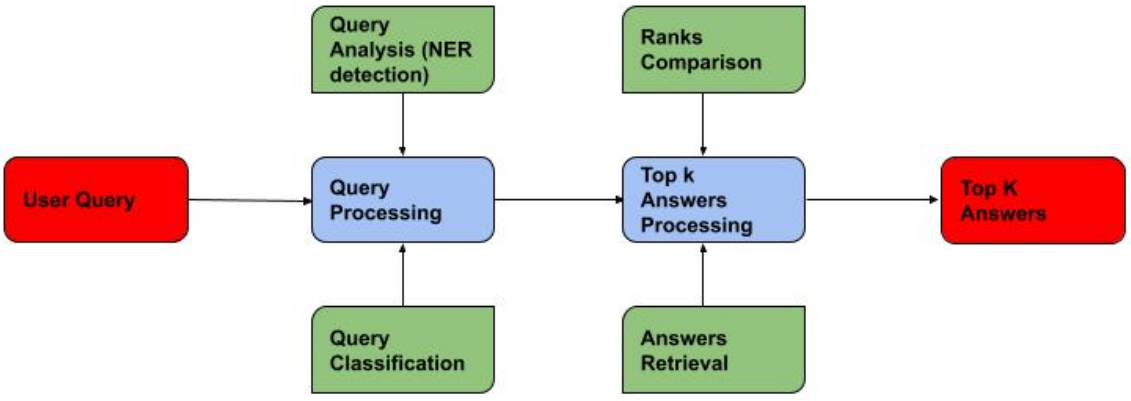

FIGURE 1.3: General architecture of QAs

For answering the queries, we first classify the query into a disease class and categories. Based on the previously answered queries in these categories, we will retrieve the top k answers for the new query. We will manually annotate the answers and compare the annotated ranks to the retrieved ranks. The overall proposed framework is shown in Figure 1.4.

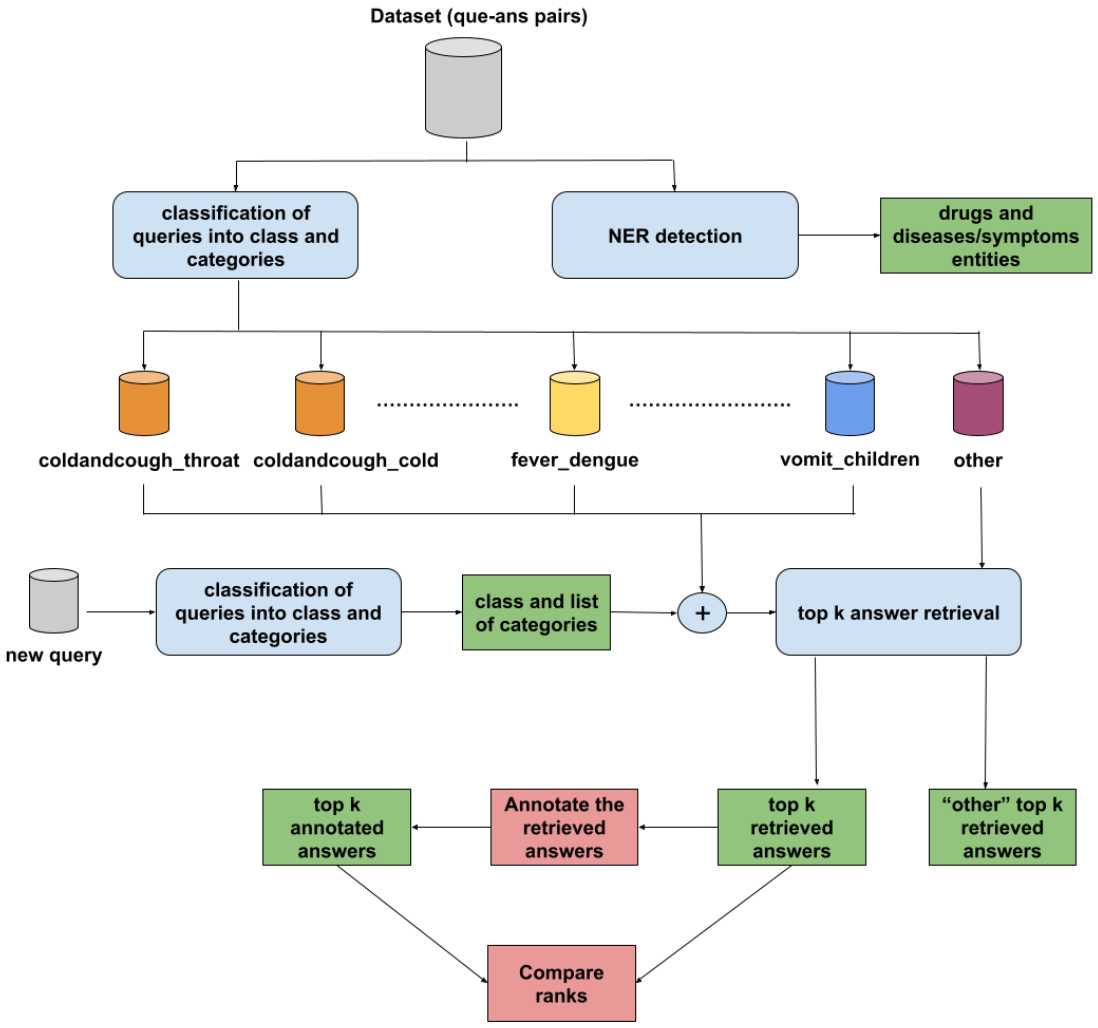

FIGURE 1.4: End-to-end flow of answering medical queries

Chapter 2

Literature Review

2.1 Medical Textual Question Answering Systems

The advent of Question Answering Systems has been envisaged as a promising solution and an efficient approach for retrieving significant information over the Internet. A considerable amount of research work has focused on open-domain QASs based on deep learning techniques due to the availability of data sources. However, the medical domain receives less attention due to the shortage of medical datasets. Therefore, in the study, the medical textual question-answering systems based on deep learning approaches were reviewed, and recent architectures of MQA systems were thoroughly explored. Furthermore, an in-depth analysis of deep learning approaches used in different MQA system tasks was provided. Finally, the different critical challenges posed by MQA systems were highlighted, and recommendations to effectively address them in forthcoming MQA systems were given out. We saw the accuracy of models varies from 70% to 90%. [3][8]

2.2 Methods for Classification

K-Means[2] is a process for partitioning an N-dimensional population into k sets on the basis of a sample. The process appears to give partitions that are reasonably efficient in the sense of within-class variance. The k-means procedure is easily programmed and computationally economical, so it is feasible to process very large samples on a digital computer. Agglomerative clustering[1] is a method for determining the mutual nearest neighbours and mutual neighbourhood value of a sample point, using the conventional nearest neighbours, is suggested. The algorithm is simple, deterministic, noniterative, requires low storage, and is able to discern spherical and nonspherical clusters. The method is applicable to a wide class of data of arbitrary shape, large size, and high dimensionality. The algorithm can discern mutually homogenous clusters. Strong or weak patterns can be discerned by properly choosing the neighbourhood width. Deep Clustering Method[7] is another algorithm optimizes the dimensionality reduction and clustering tasks jointly to substantially improve the performance. The premise behind the latter genre is that the data samples are obtained via linear transformation of latent representations that are easy to cluster; but in practice, the transformation from the latent space to the data can be more complicated.

2.3 Named Entity Recognition

Healthcare NER Models Using Language Model Pretraining[6] paper presents the approach to extracting structured information from unstructured Electronic Health Records. Their proposed NER model which was derived from scispaCy (en_ner_bc5cdr_md) uses a combination of Natural Language Processing techniques and a web-based annotation tool to optimize the performance of a custom NER model trained on a limited amount of EHR training data. Large-Scale Application of Named Entity Recognition to Biomedicine and Epidemiology[5] paper presents a python package

that can extract named entities from biomedical texts. The knowledge gained from the named entities helps the end users to see the statistics or spread of infectious disease in the least time and while parsing a large number of free texts. Named entity recognition on bio-medical literature documents using hybrid based approach[4] paper proposed a new hybrid-based approach to identify named entities from the medical literature documents. A new dictionary has been built for the route of administration, dosage forms, and symptoms to annotate the entities in the medical documents. The annotated entities are trained by the blank Spacy machine learning model. The trained model provides a decent accuracy when compared with the existing model. The hybrid model is validated with the dictionary and human (optional)to calculate the confusion matrix. It is able to identify more entities than the prevailing model. The average F1 score for five entities of the proposed hybrid based approach 73.79%.

Chapter 3

Proposed Methodology

3.1 Dataset

The data is collected from TATA's 1mg website, which is an online healthcare platform providing services like e-pharmacy, e-consultation, and health content. We collected the list of all drugs available on the website and crawled each drug's page to collect pairs of question-answer listed in the Patient queries section, related to the drug. In numbers, we collected a list of 2,77,646 drugs and collected a total of 5,94,324 queries. These are real queries asked by patients and their answers by specialized doctors on the platform. Every row consists of the following attributes:
question: query asked by the patient
answer: answer given by a specialized doctor
drug: link to drug's page on 1mg website
doctor: name of the doctor who answers the query
specialty: specialization of the doctor

The collected data consists of many repetitions of many queries. After eliminating repeating queries, the final dataset consists of 12,125 unique queries.

We annotated the first 1000 queries into the 6 defined classes. Table 3.1 shows the number of queries annotated for each class.

Class	No. of queries
cold and cough	214
fever	25
diarrhea	12
acidity	9
vomit	7
other	733

TABLE 3.1: Annotated queries for each class from the first 1000 queries

3.2 Models and Methodology

In this project, we have implemented different models for the classification of queries. We trained them and compared their accuracies and macro f1 scores. The classes for classification are: "cold and cough", "fever", "vomit", "diarrhea", "acidity" and "other". These are some of the most common diseases or symptoms in India. The "other" class includes any queries regarding disease or symptoms falling outside the rest of the five classes. Then we use a multi-output classification model on each class with different candidate labels. We classified each class into further categories, where each category represents a different cause, symptom, or behavior of the class. We did this for all classes except the "other" class. We do multi-output classification as some queries of a class may have overlapping symptoms. The classes and categories we are considering for our model are shown in Table 3.2.

Class	Categories
cold and cough	throat pain, blocked nose, runny nose, cold, cough, medication, children
acidity	digestion, headache, chest burn, nausea, medication
fever	typhoid, pain, chicken pox, dengue, medication, children
diarrhea	digestion, abdominal pain, medication, children
vomit	digestion, headache, pregnancy, medication, children
other	-

TABLE 3.2: Classes and categories for our classification models

We implemented the cosine similarity model to retrieve the answers to the queries. We have used this model to find the best possible k answers for each query from the categories they will be categorized into. We will manually annotate the ranks for the answers retrieved and compare the two ranking orders to see how the model is performing. We also find the best possible k answers from the "other" class as there may be some false positive query of the class, to see their role in answering the query. We have implemented two models for the detection of entities as drugs and symptoms/ diseases from the queries, one uses the "d4data/biomedical-ner-all" model and the second was implemented using "en_ner_bc5cdr_md" and spaCy.

3.3 Fine-tuning Methods

A good classification of queries is necessary for the effectiveness of NLP in answering medical queries. Considering the criticality, answering medical queries incorrectly can sometimes result in life-threatening situations and therefore the risk should be minimized. To get the accurate classification of medical queries, We performed fine-tuning of the models on the classification task. Fine-tuning is a commonly used technique to improve the performance of pre-trained models on a specific task. Fine-tuning involves taking a pre-trained model and adapting it to a specific task by updating its weights using task-specific data.

There are three ways in which we tried to fine-tune the model:

1. Linear Fine-tuning
2. K-shot Fine-Tuning
3. Full Fine-tuning

3.3.1 Linear fine-tuning

Linear fine-tuning is a common technique in machine learning used to adapt a pre-trained model to a specific task. The basic idea of linear fine-tuning is to take a pre-trained language model and train the topmost layer to perform a specific task. During the training process, the pre-trained weights are kept fixed, and only the weights of this layer are updated.

Linear fine-tuning is called "linear" because the top layer is typically a linear transformation of the pre-trained model's output. In text classification, the top layer is a linear layer that maps the output of the pre-trained model to a set of class labels.

Figure 3.1 shows the pictorial representation of linear fine-tuning.

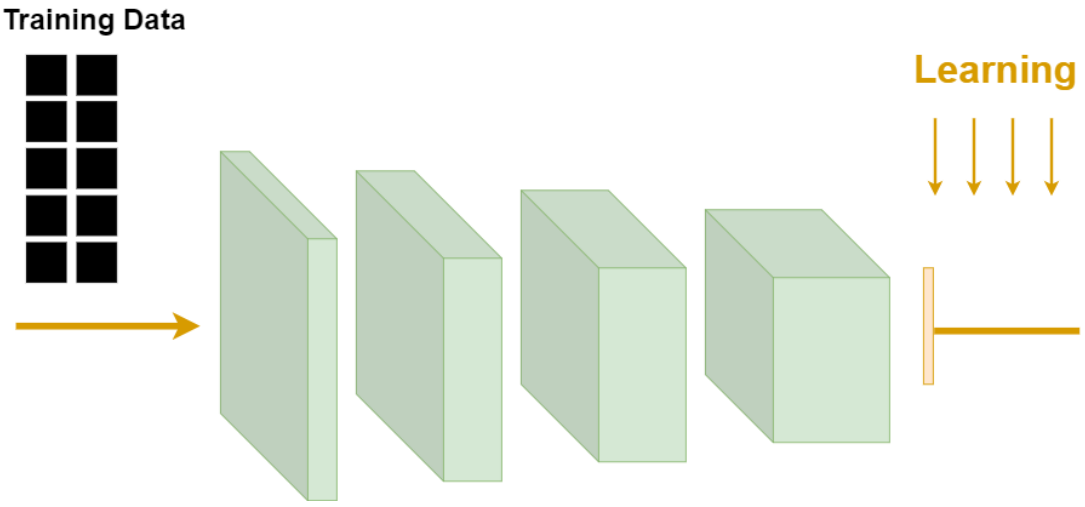

FIGURE 3.1: Linear fine tuning

3.3.2 K-shot fine-tuning

The fine-tuning process involves taking the pre-trained model and updating its parameters using the labeled data for the task. The "shot" in k-shot fine-tuning refers to the number of examples used for each update during the fine-tuning process.

For example, in 5-shot fine-tuning, the model is trained using five labeled examples from each class at each update. K-shot fine-tuning is a powerful technique for quickly adapting a pre-trained language model to a specific task while minimizing the amount of labeled data needed.

Figure 3.2 shows the pictorial representation of k-shot fine-tuning.

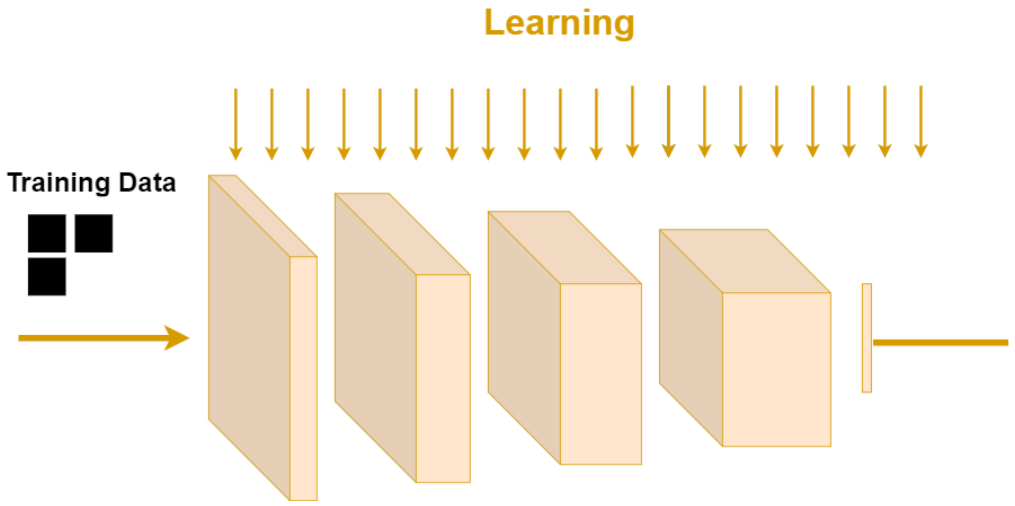

FIGURE 3.2: K-shot fine tuning

3.3.3 Full fine-tuning

Full fine-tuning is a technique used in machine learning to fine-tune an entire pre-trained model for a specific task. The basic idea of full fine-tuning is to take a pre-trained language model and continue training it on a task-specific dataset, with all of the model's weights updated during training. This means that the pre-trained weights are not frozen, as they are in linear fine-tuning. During full fine-tuning, the entire pre-trained model is trained on task-specific data, which can require a large amount of computational resources and time. Full fine-tuning can be very effective, as it allows the model to adapt not only the top layer but also the lower layers of the pre-trained model to the task-specific data. Full fine-tuning is often used when only adapting the top layer is not sufficient.

Figure 3.3 shows the pictorial representation of full fine-tuning.

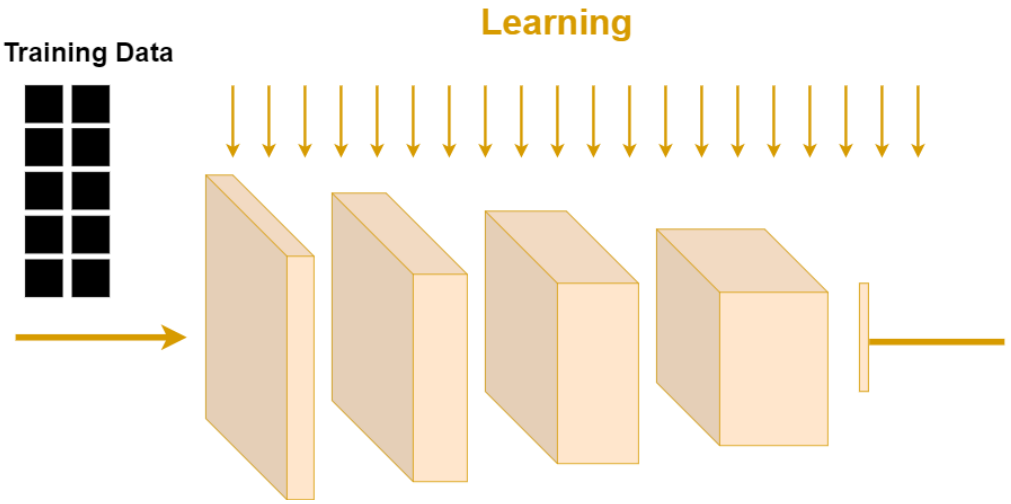

FIGURE 3.3: Full fine tuning

The choice of which fine-tuning approach to use depends on the specific task at hand and the similarity of the task to the pre-training task. Fine-tuning allows researchers to leverage the large amount of data and computational resources used for pre-training, while still achieving state-of-the-art performance on a variety of NLP tasks. We used the Hugging Face library and the PyTorch framework for implementing the fine-tuning process. We fine-tuned the models on our dataset and validated its performance on the validation set, and finally evaluated the performance on the test set. The train: validation: test split was 70:15:15.

Chapter 4

Experimental Results

4.1 Classifications

We implemented and trained different models on 80% of the data and test on the rest data. Table 4.1 shows the comparison between the macro f1-scores and accuracies of different classification models.

Model	Macro F1 score	Accuracy
Linear Regression	0.27	0.86
SVM	0.54	0.92
Decision Tree	0.67	0.90
RoBERTa large	0.57	0.82
BART large	0.70	0.89

Table 4.1: Macro f1-score and accuracy of different models

We run multi-output classification on the queries of each class to classify them into further categories. Figure 4.1 shows the classification result for some of the randomly selected queries.

Serial No.	Query	Class	Categories
1	I am getting cough from last 4 days. Pls prescribe any antibiotic. Getting coughing regularly.	cold and cough	cough, medication, throat, cold
2	What should I do for Cold and congestion, cough with green mucus clogged ear. Also I get very tired by day end.	cold and cough	cough, cold
3	Suffering from Fever for lat 3 days. Taken SINAREST tab. again Now, fever is coming. Measuring 100 in thermometer. Purchased DOLO650 & Azithral500. Any Suggestion.	fever	medication
4	Affected by Chickenpox and itchy Sensation Occurred.	fever	chicken pox
5	Which tablet to take to avoid vomiting while traveling. Thanks	vomit	medication
6	She has vomiting, head ache, sneezing, tension, thyroid problem and test showed stone. she take homeopathy medicine. But not cure. Please give correct solution.	vomit	headache, medication, digestion
7	Having telvas am for last 3-4 years. Any side effect of thismedicine. Having problem of accute acidity	acidity	medication
8	Having problem of acid reflux disease with throat infection and pain in right side of abdomen	acidity	digestion
9	Dear Doctor i feel constipation while passing the stool in the morning i even feel pain after the motion the pain occur after 1 to 2 hours of motion. I even have to visit the toilet 2 to 3 times to clear the stomach .Please help me in this regard.	diarrhea	digestion, abdominal pain
10	3years old baby girl is taking augmentin duo & enterogemina but still suffering pain with frequent stool.Having haemoglobin is 9.6.	diarrhea	children, medication, digestion
11	Mixed anxiety depressive disorder	other	-
12	Parkinson's disease tremors ,Insomnia.	other	-

FIGURE 4.1: Classification result for some queries

We apply the classification model to the rest of the 11,125 queries and selected 30 queries from every five classes (except the "other" class) randomly, giving us a total of 150 queries. We manually annotate their class to calculate the accuracy for each class, Table 4.2 below illustrates the class-wise accuracies for these 150 queries.

Class	Accuracy
cold and cough	90.0%
fever	90.0%
diarrhea	66.6%
acidity	46.6%
vomit	90.0%
total	76.6%

TABLE 4.2: Class-wise accuracies of 150 random selected queries

The accuracy was not good for the "diarrhea" and "acidity" classes. Upon manual inspection, we find out that model classified some queries related to diabetes and

stomach aches into diarrhea and acidity respectively.

4.2 Answers generation

We classified the above 150 queries into categories by running the multi-label classification. We used the first 1000 queries as our corpus for retrieving answers to these 150 queries and retrieved the best k=3 answers from their respective categories. We annotate ranks for the answers retrieved based on how significant the answer is for the query and compared them with the retrieved ranks. We also retrieved the answers from the "other" class.

Out of 150 queries:
1. For 20 queries, the answers were unsatisfactory
2. For 24 queries, the answers retrieved from the "other" class were better than the answers retrieved from its own class and categories
(34 out of these 44 queries were false positives in their respective classes)
3. For the rest of the 106 queries, the answers retrieved were satisfactory

Figure 4.2 shows the comparison between the retrieved rank(rank output from the model) vs Annotated rank(manual rank) of top k=3 answers for the 106 queries with satisfactory answers. The first group of bars on the x-axis, all have 1 as their retrieved rank. The blue bar of this group represents no. of queries with Annotated rank 1, the red bar represents no. of queries with Annotated rank 2, and the yellow bar represents no. of queries with Annotated rank 3. The same color rank convection follows for the second group of bars having retrieved rank 2 and the last group of bars having retrieved rank 3.

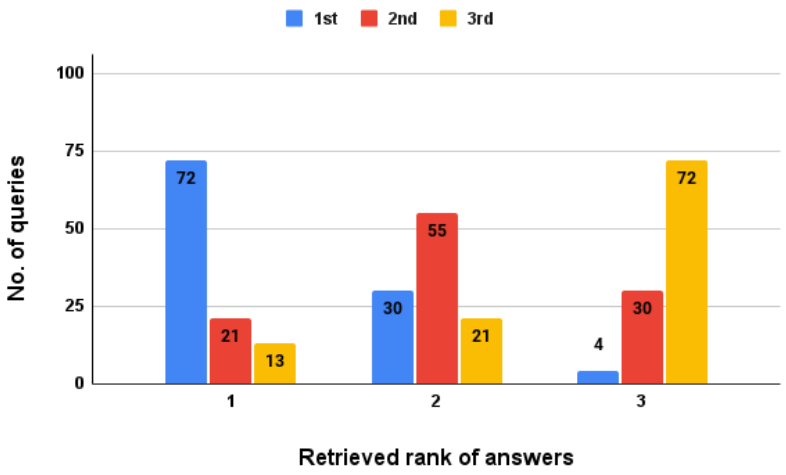

FIGURE 4.2: Comparison between retrieved ranks and annotated ranks of top k=3 answers

Out of 150 selected queries, one query was "Medicine for throat pain and cold". Table 4.3 shows the classification scores for the query and Table 4.4 shows the multi-label classification scores for the query:

Class	Scores
cold and cough	87.1%
fever	7.6%
diarrhea	2.5%
acidity	1.6%
vomit	0.8%
other	0.4%

TABLE 4.3: Classification scores for the query "Medicine for throat pain and cold". Based on these probabilistic scores, the query is classified into the "cold and cough" class.

Class	Scores
medication	99.8%
throat pain	99.2%
cough	93.4%
cold	65.8%
blocked nose	12.1%
runny nose	3.7%
children	0.7%

TABLE 4.4: Multi-output classification scores the for query "Medicine for throat pain and cold". Based on these probabilistic scores, it is categorized into "medication", "throat pain" and "cough"(we set a standard threshold frequency of 0.80).

We retrieved answers for k=3 to the query in the categories "medication, "throat pain" and "cold". Annotated rank order for retrieved answers is 1-2-3, which is the same as the retrieved ranks.

```
Concern:  Medicine for throat pain and cold
Ans1:  Tab zerodol P BDBetadine gargles for throat
Ans2:  Tab Dolo 650mg twice daily Alex cough syp twice daily
Ans3:  Take dolo 650 mg three times a day for two days, tab alaspan am once a day for five days
```

FIGURE 4.3: Top k=3 retrieved answers for the query "Medicine for throat pain and cold"

4.3 Fine-tuning

The best obtained F1 score and accuracy were 0.7 and 89% respectively, so, we performed fine-tuning of the BERT model on the classification task to better the performance. We extended the annotation of data up to 3000 queries. Table 4.5 below shows the number of queries annotated for each class.

Class	No. of queries
cold and cough	350
fever	88
diarrhea	39
acidity	45
vomit	55
other	2423

TABLE 4.5: Annotated queries for each class from the first 3000 queries

4.3.1 Linear fine-tuning

We implemented and fine-tuned our BERT model using the linear fine-tuning method to preserve the learning of all layers of the BERT model except the linear layer (and a few last transformers layers). Linear layers apply a linear transformation to the

input data, followed by a bias term. We kept the original values of hyperparameters and trained the last layer (or the last few layers) and compared the macro F1 scores. The F1 score is a commonly used metric to evaluate the accuracy of a model, particularly in the case of an imbalanced dataset. Table 4.6 below compares the macro F1 score for different possibilities of linear fine-tuning.

Linear fine-tuning	Macro F1 score
linear layer	0.15
last 2 transformers layers and linear layer	0.41
last 3 transformers layers and linear layer	0.40
last 4 transformers layers and linear layer	0.35

TABLE 4.6: Macro F1 score for different possibilities of linear fine-tuning

4.3.2 K-shot fine-tuning

We fine-tuned our model with different values of k starting from 1, since the "diarrhea" class contains the least number of queries (39), and after doing train: validation: test split, we are constrained with 25 as the maximum value for the k. Table 4.7 below compares the macro F1 score for different values of k and weight_decay.

k_value	weight_decay=0.01	weight_decay=0.001
k=1	0.06	0.01
k=5	0.12	0.07
k=10	0.15	0.16
k=15	0.29	0.33
k=20	0.37	0.34
k=25	0.33	0.39

TABLE 4.7: Macro F1 score for different values of k and weight_decay in k-shot fine-tuning

4.3.3 Full fine-tuning

We varied the learning rate and weight decay hyperparameters to evaluate their effect on the F1 score of a fully fine-tuned BERT model. All possible combinations of the learning rate and weight decay values were tested, and the F1 score was calculated for each combination. This allowed us to identify the combinations that produced the best performance for the specific task being evaluated. Table 4.8 below compares the macro F1-score for different values of learning rate and weight decay.

learning_rate	weight_decay=0.01	weight_decay=0.001
learning_rate=1e-6	0.28	0.27
learning_rate=1e-5	0.83	0.87
learning_rate=1e-4	0.15	0.15
learning_rate=1e-3	0.15	0.15

TABLE 4.8: Macro F1 score for different values of learning rate and weight_decay in full fine-tuning

This experiment highlights the importance of carefully tuning hyperparameters during full fine-tuning to achieve the best possible performance. The results of the experiment showed that the optimal combination of learning rate = 10^{-5} and weight decay = 0.001 gives the best Macro F1 score of 0.87 and the corresponding accuracy is 95%. An increase of 0.17 in Macro F1 score from before.

We also experimented with adding class_weights to the model with hyperparameters learning_rate = 10^{-5} and weight_decay = 0.001 and fully fine-tune the model. Class_weights is a way to balance the influence of different classes during training. It involves assigning a weight to each class, where the weight is inversely proportional to the frequency of samples in the class. This means that classes with fewer samples are assigned a higher weight, while classes with more samples are assigned a lower weight. The macro F1 score and accuracy dropped to 0.78 and 95%, however, better than many others.

Similarly, we fully fine-tuned BART and RoBERTa models and obtained Macro F1 scores of 0.8 and 0.77 respectively. Figure 4.4 below demonstrates how the Macro F1 score and Accuracy are employed to evaluate different classification models. X-axis represents the classification models along with the type of fine-tuning applied, for example, the BERT(K-shot) means the best BERT model performance after K-shot fine-tuning.

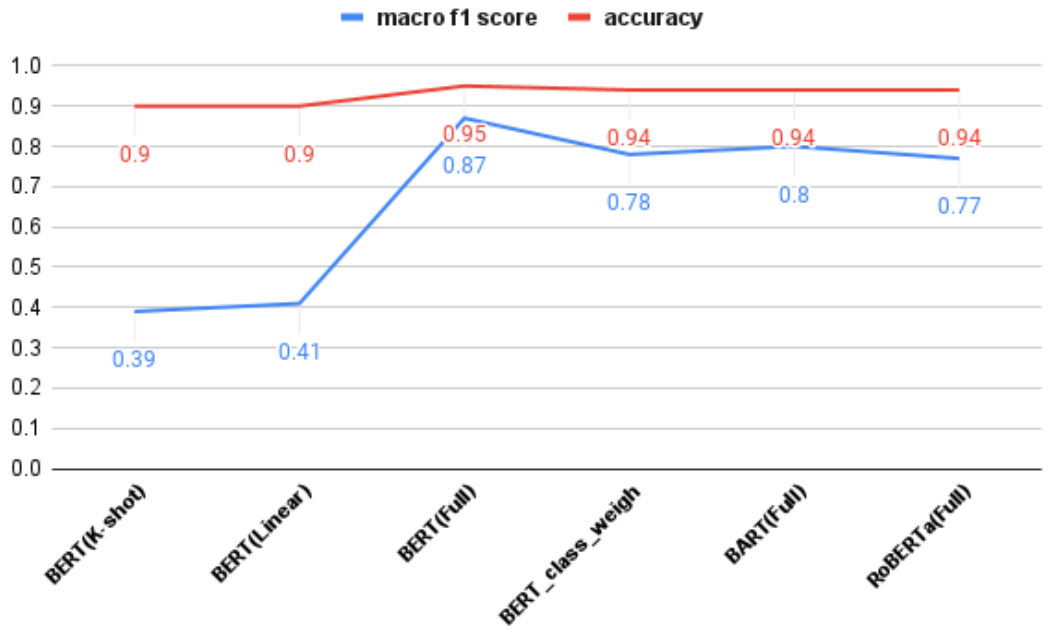

FIGURE 4.4: The evaluation of different classification models

4.4 NER detection

We implemented two models for NER detection: "d4data/biomedical-ner-all" [5] and "en_ner_bc5cdr_md"[6]. These models were detecting entities into two categories: drugs and diseases/symptoms, "d4data/biomedical-ner-all" detects entities as 'Medication' and 'Sign_symptom' whereas "en_ner_bc5cdr_md" detects entities as 'CHEMICAL' and 'DISEASE'. We applied both models to the first 100 queries and calculate class-wise accuracies.

Table 4.9 compares the class-wise accuracies of the NER results of the two models.

class	d4data/biomedical-ner-all	en_ner_bc5cdr_md
cold and cough	0.43	0.43
fever	0.52	0.52
vomit	0.50	0.50
diarrhea	0.50	0.25
acidity	0.29	0.13
other	0.41	0.53

TABLE 4.9: Class-wise accuracies of two NER models

One of the queries was "Can we give ambrodil-s and polymol kid for 6 months baby at a time for fever and cough?"

Table 4.10 shows the detection of drugs and diseases/symptoms entities for the above query using two models.

value	d4data/biomedical-ner-all	en_ner_bc5cdr_md
ambrodil-s	Medication	-
polymol	Medication	-
fever	Sign_symptom	DISEASE
cough	Sign_symptom	DISEASE

TABLE 4.10: Result of NER detection for the query "Can we give ambrodil-s and polymol kid for 6 months baby at a time for fever and cough?"

The "d4data/biomedical-ner-all" model was able to detect diseases properly, but for the drugs, if the drug name comprises more than 1 word, it only detects the first word as "Medication" and ignores the following words.

For example: one of the drug names is "Augmentin DUO", the implemented model only detected Augmentin as medication ignoring the rest part of the drug name.

Chapter 5

Conclusion and Future Works

5.1 Conclusion

1. Worked with various classification models and from the results, the full fine-tuning "BERT" model provides the best accuracy in classifying the medical queries into classes of diseases.

2. The best overall performance of the final model after fine-tuning was good enough keeping the criticality of the situation in mind. We are able to accomplish 95% accuracy with a Macro F1 score of 0.87.

3. Concluded that adding class_weights to the pre-trained model results in relatively poor performance than fine-tuning the model without them.

4. On the basis of the results, we concluded that fine-tuned BERT model works better than other models for imbalanced datasets.

5. On the basis of the class and categories a query is classified into, we are able to automatically retrieve answers for the query asked. We concluded that the answers were satisfactory if the query is classified accurately.

6. Medical QA has made significant progress in recent years due to the use of NLP and deep learning techniques in this area. Automatic QA has been possible in many medical question–answering systems, and the availability of corpus data in the medical domain is increasing over time.

5.2 Future Works

1. Improve the NER models to get more accurate detection of drugs and diseases/symptoms. Especially for drugs with multiple words in its name, currently, models are only able to detect the first word of the drugs and not detecting the rest part of the name.

Bibliography

[1] K. Chidananda Gowda and G. Krishna. Agglomerative clustering using the concept of mutual nearest neighbourhood. *Pattern Recognit.*, 10:105–112, 1978.

[2] J.B MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics, University of California Press, Berkeley*, pages 281–297, 1967.

[3] Emmanuel Mutabazi, Jianjun Ni, Guangyi Tang, and Weidong Cao. A review on medical textual question answering systems based on deep learning approaches. *Applied Sciences*, 11(12), 2021.

[4] R Ramachandran and K Arutchelvan. Named entity recognition on bio-medical literature documents using hybrid based approach. *Journal of Ambient Intelligence and Humanized Computing*, 2021.

[5] Shaina Raza, Deepak John Reji, Femi Shajan, and Syed Raza Bashir. Large-scale application of named entity recognition to biomedicine and epidemiology. *medRxiv*, 2022.

[6] Amogh Kamat Tarcar, Aashis Tiwari, Vineet Naique Dhaimodker, Penjo Rebelo, Rahul Desai, and Dattaraj Rao. Healthcare ner models using language model pretraining, 2020.

[7] Bo Yang, Xiao Fu, Nicholas D Sidiropoulos, and Mingyi Hong. Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In *international conference on machine learning*, pages 3861–3870. PMLR, 2017.

[8] Shuohua Zhou and Yanping Zhang. Datlmedqa: A data augmentation and transfer learning based solution for medical question answering. *Applied Sciences*, 11(23), 2021.