

# **MEDICAL DIAGNOSIS USING NATURAL LANGUAGE PROCESSING**

**A PROJECT REPORT**

*Submitted by*

**AKSHAYA . S** **910020104004**

**AKSHITHA . V . A** **910020104005**

**KEERTHI . P** **910020104020**

*in partial fulfillment for the award of the degree*

*of*

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

**ANNA UNIVERSITY REGIONAL CAMPUS, MADURAI**



**ANNA UNIVERSITY : CHENNAI 600 025**

**MAY 2023**

# **MEDICAL DIAGNOSIS USING NATURAL LANGUAGE PROCESSING**

**A PROJECT REPORT**

*Submitted by*

**AKSHAYA . S** **910020104004**

**AKSHITHA . V . A** **910020104005**

**KEERTHI . P** **910020104020**

*in partial fulfillment for the award of the degree*

*of*

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

**ANNA UNIVERSITY REGIONAL CAMPUS, MADURAI**



**ANNA UNIVERSITY : CHENNAI 600 025**

**MAY 2023**

**ANNA UNIVERSITY : CHENNAI 600 025**

**BONAFIDE CERTIFICATE**

Certified that this project report “**MEDICAL DIAGNOSIS USING NATURAL LANGUAGE PROCESSING**” is the bonafide work of **AKSHAYA S (910020104004), AKSHITHA V A (910020104005), KEERTHI P (910020104020)**, who carried out the project work under my supervision.

**SIGNATURE**

**Dr.E.Srie Vidhya Janani**

**SUPERVISOR**

Assistant Professor,  
Department of Computer Science and  
Engineering,  
Anna University Regional Campus,  
Madurai - 625019

**SIGNATURE**

**Dr.E.Srie Vidhya Janani**

**HEAD OF THE DEPARTMENT**

Assistant Professor,  
Department of Computer Science and  
Engineering,  
Anna University Regional Campus,  
Madurai – 625019

Submitted for the University Examination held on \_\_\_\_\_

**INTERNAL EXAMINER**

**EXTERNAL EXAMINER**

# ACKNOWLEDGEMENT

We would like to thank our dean **Dr.K.Linga Durai**, for providing infrastructure facilities and whole hearted encouragement for completing our project successfully.

We pay our grateful acknowledgement and extend our sincere gratitude to **Dr.E.Srie Vidhya Janani, M.E., Ph.D.**, Head of the Department, Computer Science and Engineering, Anna University Regional Campus, Madurai, Anna University, Chennai, for extending the facilities of the department towards our project and for her unstinting support.

We express our thanks to our guide, **Dr.E.Srie Vidhya Janani, M.E., Ph.D.**, Assistant Professor, Anna University Regional Campus, Madurai, Anna University, Chennai, for guiding us through every phase of the project. We appreciate her thoroughness, tolerance and ability to share knowledge with us. We thank her for being easily approachable and quite thoughtful. We owe her for harnessing our potential and bringing out the best in us. We appreciate her immense support.

We would like to thank our Class Advisor **Dr.P.Uma Maheswari, MCA., Ph.D.**, and all our teaching and non-teaching faculty members of the Department and also our fellow friends for helping us in providing valuable suggestions and timely ideas for the successful completion of the project. We extend our thanks to our family members who have been a great source of inspiration and strength to us during the course of this Project work. We sincerely thank all of them.

**AKSHAYA S**

**AKSHITHA V A**

**KEERTHI P**

# ABSTRACT

Healthcare is vital for a good life. However, it's difficult to get immediate consultation from the right doctor in case of any health issues. Identifying the right doctor to consult is often lacking in time and cost effectiveness, particularly for the patient. The project “Medical Diagnosis using Natural Language Processing” aims to design a medical diagnosis system which will detect the disease based on the patients’ symptoms and suggest the right healthcare professional to consult. The corpus used is formed from trusted textbooks used in the medical field. Natural Language Processing is used to extract and pre-process data from these textbooks. When the user inputs the observed symptoms to the system, the system will test perform cosine similarity using techniques such as CountVectorizer and TF-IDFVectorizer to compare it with the knowledge base. A logistic Regression model is implemented to test the probability of match against each type of diseases and finally suggest the specialist to visit. A simple and easy to use user interface has also been designed for this project. The ultimate aim is to provide a cost and time effective automated system that acts as an assistant to healthcare professionals in diagnosing diseases.

**Keywords:** Medical Diagnosis, Natural Language Processing, Cosine Similarity, Logistic Regression.

# TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	<b>ABSTRACT</b>	<b>ii</b>
	<b>LIST OF TABLES</b>	<b>vii</b>
	<b>LIST OF FIGURES</b>	<b>viii</b>
	<b>LIST OF ABBREVIATIONS</b>	<b>ix</b>
<b>1</b>	<b>INTRODUCTION</b>	<b>9</b>
	1.1 OVERVIEW	9
	1.2 PROBLEM DESCRIPTION	11
	1.3 AIM AND OBJECTIVE	11
<b>2</b>	<b>LITERATURE REVIEW</b>	<b>13</b>
<b>3</b>	<b>SYSTEM SPECIFICATION</b>	<b>15</b>
	3.1 REQUIREMENTS ANALYSIS	15
	3.1.1 FUNCTIONAL REQUIREMENTS	15
	3.1.2 NON-FUNCTIONAL REQUIREMENTS	15
	3.1.3 DATA SOURCE	15
	3.1.4 OPERATING ENVIRONMENT	15
	3.1.5 DESIGN AND IMPLEMENTATION	16
	3.1.6 ASSUMPTIONS AND DEPENDENCIES	16
	3.2 HARDWARE REQUIREMENTS	16
	3.3 SOFTWARE REQUIREMENTS	16

	3.4 SOFTWARE DESCRIPTION	16
	3.4.1 PYTHON 3.7.4	16
	3.4.2 NUMPY	17
	3.4.3 PANDAS	18
	3.4.4 NLTK	18
	3.4.5 SKLEARN	19
	3.4.6 GENSIM	19
	3.4.7 COREX	19
	3.5 SOFTWARE QUALITY ATTRIBUTES	20
<b>4</b>	<b>PROPOSED SYSTEM</b>	<b>21</b>
	4.1 EXISTING SYSTEM	21
	4.1.1 DISADVANTAGES	21
	4.2 PROPOSED SYSTEM	21
	4.2.1 DATA EXTRACTION AND PRE-PROCESSING	21
	4.2.2 TOPIC MODELLING	23
	4.2.3 COSINE SIMILARITY	24
	4.2.4 CLASSIFICATION AND PREDICTION	24
	4.4 ADVANTAGES	24
<b>5</b>	<b>ARCHITECTURE AND DESIGN</b>	<b>26</b>
	5.1 SYSTEM ARCHITECTURE	26
	5.2 DATA FLOW DIAGRAM	28
	5.3 USE CASE DIAGRAM	29
	5.4 SEQUENCE DIAGRAM	30
	5.5 CLASS DIAGRAM	31
	5.6 INTERACTION DIAGRAM	32
	5.7 STATE OR ACTIVITY DIAGRAM	32
	5.8 COMPONENT AND DEPLOYMENT	34

	DIAGRAM	
	5.9 ER DIAGRAM	34
<b>6</b>	<b>PERFORMANCE ANALYSIS</b>	<b>36</b>
	6.1 UNIT TESTING	36
	6.2 INTEGRATION TESTING	37
	6.3 USER TESTING	37
	6.4 RESULT	38
	6.5 RESULT ANALYSIS	39
<b>7</b>	<b>CONCLUSION AND FUTURE SCOPE</b>	<b>40</b>
	7.1 CONCLUSION	40
	7.2 FUTURE SCOPE	41
<b>8</b>	<b>APPENDICES</b>	<b>41</b>
	8.1 SAMPLE CODE	41
	8.2 SCREENSHOTS	45
<b>9</b>	<b>REFERENCES</b>	<b>47</b>



## **LIST OF TABLES**

<b>TABLE NO.</b>	<b>TABLE NAME</b>	<b>PAGE NO.</b>
<b>6.1</b>	SAMPLE UNIT TESTING	<b>37</b>
<b>6.4</b>	INPUTS AND PREDICTED RESULTS	<b>38</b>
<b>6.5</b>	PREDICTED RESULTS COMPARISION	<b>39</b>

## **LIST OF FIGURES**

<b>FIGURES NO.</b>	<b>DESCRIPTION OF FIGURE</b>	<b>PAGE NO.</b>
4.1	DATA SOURCE	22
4.2	CORPUS	23
4.3	DOCTORS AND DESCRIPTION	23
4.4	COSINE CV AND TF-IDF	24
5.1	SEQUENTIAL MODEL	26
5.2	DATA FLOW DIAGRAM	28
5.3	USE CASE DIAGRAM	29
5.4	SEQUENCE DIAGRAM	30
5.5	CLASS DIAGRAM	31
5.6	INTERACTION DIAGRAM	32
5.7	STATE OR ACTIVITY DIAGRAM	33
5.8	COMPONENT AND DEPLOYMENT DIAGRAM	34
5.9	ENTITY – RELATIONSHIP DIAGRAM	34

## **LIST OF ABBREVIATIONS**

NLP	Natural Language Processing
COREX	Correlation Explanation
NLTK	Natural Language Toolkit
LDA	Latent Dirichlet Allocation
CSV	Comma Separated Values
TF-IDF	Term Frequency-Inverse Document Frequency
CV	Count Vectorizer

# **CHAPTER 1**

## **INTRODUCTION**

### **1.1 OVERVIEW**

Medical diagnosis systems are becoming increasingly popular and accurate, with enormous advantages such as cost-effectiveness, speed and reliable decision support for diagnostics of disease, illness, injury, and other physical and mental damages in human beings. The rise in remote health services (or tele health) offered by healthcare institutions coincided with the evolution of assisted living systems and environments, aiming to widen the possibility for older and disadvantaged people to access appropriate healthcare services. With the increase in the innovation of medical technologies, there is a need to adopt medical expert systems that will oversee and control diagnosis.

The development of the medical domain-oriented diagnosis systems has been addressed by several researchers. Such systems powered by AI techniques may serve patients with minor health concerns while allowing medical doctors to predict the disease in a very short span of time. Medical diagnostic processes carried out with the aid of computer-related technology which is on the rise have improved the experience and capabilities of physicians to make an effective diagnosis of diseases while employing novel signal processing techniques for analysis of patients' physiological data. With the rise of the artificial intelligence (AI) techniques, the medical diagnosis system has appeared as a promising direction in streamlining the communication between doctors and patients. Such systems are becoming increasingly popular as remote health interventions are implemented in the form of the synchronous text-based diagnosis systems. Patients with chronic diseases could make the most advantage from the use of

medical diagnosis system which can monitor their condition, provide accurate information. For the effective use of medical diagnosis system in the healthcare domain, the medical diagnosis system technology needs advanced reasoning capabilities based on the formalization of medical knowledge (semantics) and health state of patients coupled with language vocabularies.

The natural language processing (NLP) technology can serve as a tool between computers and humans using linguistic analysis to obtain knowledge from an unstructured free text. The NLP systems have shown their uniqueness and importance in the areas of information retrieval mostly in the retrieval and processing of large amount of unstructured clinical records and return structured information by user-defined queries. In general, the NLP system is aimed at representing explicitly the knowledge that is expressed by the text written in a natural language. There are few applications of the NLP techniques in diagnosing diseases despite the enormous amount of text-based information, which can be retrieved from patients' self-narrations. Linguistic structures such as co references make medical texts difficult to be interpreted. Moreover, unique linguistic entities such as medical abbreviations make the inference of knowledge from medical texts much harder.

This study introduces the use of the NLP model to diagnose the acquired disease. The application of NLP techniques to assist medical experts in their diagnosis would serve as a boost in successfully improving healthcare services through effective analysis of narrative text of symptoms provided by a patient, and by the use of medical textbooks as a dataset. The focus of the study was based on medical diagnosis to show the relationship of information retrieval and text mining to the medical diagnosis problem. The study concludes that the proposed system would help in improving the goals of providing a effective medical diagnosis. Many concluded that the proposed system clinically gave a

better accuracy and speed, thereby improving the efficiency and quality of service.

## **1.2 PROBLEM DESCRIPTION**

The healthcare industry is one among the most important sectors across the world- both economically and employment-wise, making it one among the busiest industries within the world. With this height of the chaos, demand for speed and efficiency is important for contemporary patients who want quick and straightforward access to information.

## **1.3 AIM AND OBJECTIVE**

Medical Diagnosis System has the potential to revolutionize healthcare. The aim of the project “Medical Diagnosis System using Natural Language Processing” is to develop an automated system that will predict the type of disease a person might have from their symptoms and suggest the health specialist that they should consult. This will substantially boost efficiency and improve the accuracy of symptoms collection and ailment identification. With the technological advancements of AI, medical diagnosis system has begun to be an excellent tool for quick and straightforward automation. Now you don’t get to hold in line for hours, before a representative invests time to seem into your symptoms, while a Medical Diagnosis System can do that instantly.

The objectives of this project are:

- To extract data from medical textbooks
- To pre-process the extracted data by removing stop words, tokenization, lemmatization, removing special characters and digits, and other NLP techniques that will result in a clean data source.
- To perform topic modelling using LSA, LDA, and CorEx
- To implement Logistic Regression for classification

Overall, this medical diagnosis system can promptly respond to the problems that arise in daily life and to the health state changes of people who might otherwise have to spend exorbitant time and money to identify the specialist that should be consulted.

## **CHAPTER 2**

### **LITERATURE REVIEW**

**Title 1 :** Text Messaging-Based Medical Diagnosis Using Natural Language Processing and Fuzzy Logic (Hindawi Journal of Healthcare Engineering volume 2020)

**Authors:** Nicholas A. I. Omoregbe, Israel O. Ndaman, Sanjay Misra, Olusola O Abayomi

**Methodology:** The study assesses the clinical data needs and requirements in diagnosing the tropical diseases in Nigeria and assesses the patients' clinical data found in EHRs or manual records. The proposed text-based medical diagnosis system are as follows: (1) description of the knowledge base; (2) pre-processing of text-based documents; (3) tagging of document; (4) extraction of answer; (5) ranking of candidate answers

**Limitations:** Lack of automation of this medical diagnosis system to easily recognize diseases. No Audio interaction in- corporate to make the system more interactive.

**Title 2 :** Joint POS Tagging and Dependency Parsing with Transition based Neural Networks (2019)

**Authors:** Liner Yang, Meishan Zhang, Yang Liu, Maosong Sun, Nan Yu, Guohong Fu

**Methodology:** In this method, the system database maintains a dataset of synonyms for important keywords in that domain. The sentence sent by the user is then mapped on to that synonym dataset. The keywords detected from the



sentence are then checked in that synonym set to check for same intent. All possible synonyms of that keyword are then looked out for a match in the main database. The sentence which is closest to the user sentence is extracted.

**Limitations:** This method is time consuming and requires more of storage and complexity.

## **CHAPTER 3**

### **SYSTEM SPECIFICATION**

#### **3.1 REQUIREMENTS ANALYSIS**

##### **3.1.1 Functional Requirements**

**Collecting Dataset:** Creating a dataset for the training of the models is the primary task. Datasets can be created manually and then augmented.

**Pre-Processing:** Sending the dataset to the model, the dataset should be pre-processed. This includes splitting the data into three folds.

**Training Models:** To do an optimal phishing detection model, they should be trained and validated.

##### **3.1.2 Non-Functional Requirements**

- Accuracy
- Less training time
- Simple architecture

##### **3.1.3 Data Source**

The primary data source is a medical textbook or disease handbook. In our project, Professional Guide to Diseases by Laura Wilkins is used

##### **3.1.4 Operating Environment:**

As of now this application is only available for PC users having operation system Windows 7 and above.

### **3.1.5 Design and Implementation:**

We have designed our Medical Diagnosis System using Python. We have used various python libraries like numpy, pandas, sklearn to pre-process the data, creating training data and testing data.

### **3.1.6 Assumptions and Dependencies**

We have optimised our Medical Diagnosis System in such a way that it uses very less RAM, which makes our application very flexible across all sets of devices.

## **3.2 HARDWARE REQUIREMENTS**

The hardware requirements for this product are:

- **System:** Intel Pentium
- **Hard disk:** 120 GB
- **RAM:** 2 GB

## **3.3 SOFTWARE REQUIREMENTS**

The software requirements for this product are:

- **Operating system:** Windows 7 and above
- **Coding Language:** Python 3.x
- **Libraries:** NLTK, SKLEARN, COREX, NYMPY, PANDAS, SPACY, GENSIM

## **3.4 SOFTWARE DESCRIPTION**

### **3.4.1 Python 3.7.4**

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. The biggest strength of Python is huge collection of standard libraries which can be used for the following:

- Machine Learning
- GUI Applications (like Kivy, Tkinter, PyQt etc.)
- Web frameworks like Django (used by YouTube, Instagram, Dropbox)
- Image processing (like OpenCV, Pillow)
- Web scraping (like Scrapy, BeautifulSoup, Selenium)
- Test frameworks
- Multimedia Scientific computing
- Text processing and many more.

### 3.4.2 NumPy

NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python. It is open-source software. It contains various features including these important ones:

- A powerful N-dimensional array object
- Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier transform, and random number capabilities

Besides its scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data.

### 3.4.3 Pandas

Pandas is an open-source library that is made mainly for working with relational or labelled data both easily and intuitively. It provides various data structures and operations for manipulating numerical data and time series. This library is built on top of the NumPy library. Pandas is fast and it has high performance & productivity for users.

It provides:

- Fast and efficient for manipulating and analysing data.
- Data from different file objects can be loaded.
- Easy handling of missing data (represented as NaN) in floating point as well as non-floating point data
- Size mutability: columns can be inserted and deleted from DataFrame and higher dimensional objects
- Data set merging and joining.
- Flexible reshaping and pivoting of data sets
- Provides time-series functionality.
- Powerful group by functionality for performing split-apply-combine operations on data sets.

### 3.4.4 NLTK

The Natural Language Toolkit, or more commonly NLTK, is a suite of libraries and programs for symbolic and statistical natural language processing (NLP) for English written in the Python programming language. It supports classification, tokenization, stemming, tagging, parsing, and semantic reasoning functionalities. It was developed by Steven Bird and Edward Loper in the Department of Computer and Information Science at the University of Pennsylvania. NLTK includes graphical demonstrations and sample data.

NLTK is intended to support research and teaching in NLP or closely related areas, including empirical linguistics, cognitive science, artificial intelligence, information retrieval, and machine learning. NLTK has been used successfully as a teaching tool, as an individual study tool, and as a platform for prototyping and building research systems.

### **3.4.5 sklearn**

scikit-learn (formerly scikits.learn and also known as sklearn) is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support-vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

### **3.4.6 Gensim**

Gensim is an open-source library for unsupervised topic modelling, document indexing, retrieval by similarity, and other natural language processing functionalities, using modern statistical machine learning. It is designed to handle large text collections using data streaming and incremental online algorithms, which differentiates it from most other machine learning software packages that target only in-memory processing.

Gensim includes streamed parallelized implementations of fastText, word2vec and doc2vec algorithms, as well as latent semantic analysis (LSA, LSI, SVD), non-negative matrix factorization (NMF), latent Dirichlet allocation (LDA), tf-idf and random projections.

### **3.4.7 CorEx**

Correlation Explanation (CorEx) is a topic model that yields rich topics that are maximally informative about a set of documents. The advantage of using

CorEx versus other topic models is that it can be easily run as an unsupervised, semi-supervised, or hierarchical topic model depending on a user's needs. For semi-supervision, CorEx allows a user to integrate their domain knowledge via "anchor words." This integration is flexible and allows the user to guide the topic model in the direction of those words. This allows for creative strategies that promote topic representation, separability, and aspects. More generally, this implementation of CorEx is good for clustering any sparse binary data.

### **3.5 SOFTWARE QUALITY ATTRIBUTES**

- **Adaptability** - The product is very adaptable. It can work on maximum platforms; its features might change with change of platform.
- **Availability** - This product will be available anywhere.

## **CHAPTER 4**

### **PROPOSED SYSTEM**

#### **4.1 EXISTING SYSTEM**

The existing system implements a Question Answering System which can be identified as an information accessing system that tries to answer natural language queries by giving suitable answers making use of attributes available in natural techniques.

This system takes plain text as input and gives qualified answer as the output.

##### **4.1.1 DISADVANTAGES**

The disadvantages in existing system are:

- Difficult to extract knowledge from the medical crowd-sourced Q&A websites.
- Irrelevant question-answer pairs may be extracted.
- The questions asked by patients may be ambiguous.

#### **4.2 PROPOSED SYSTEM**

The proposed system is an automated system that will predict the type of disease a person might have from their symptoms and suggest the health specialist that they should consult. This takes place in several steps:

##### **4.2.1 Data Extraction and Pre-processing**

In order to achieve the specified objective, a trusted medical textbook is used, that is, “Professional Guide to Diseases” by Laura Wilkins. The cover is shown in figure 3.1. This book is universally accepted and used by



medical students. The PDF version of this book is used. Each chapter of the book groups diseases by the body system affected. The required data is extracted using NLP techniques such as tokenization, lemmatization, POS tagging and such.

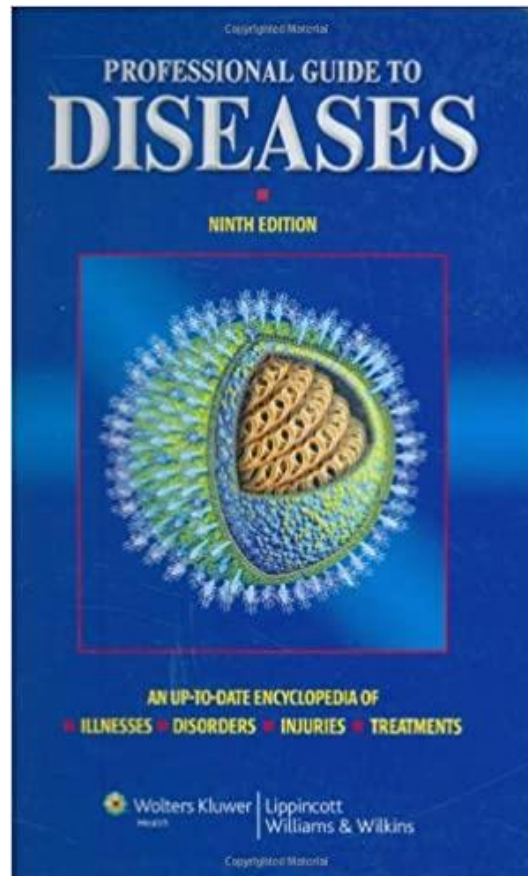


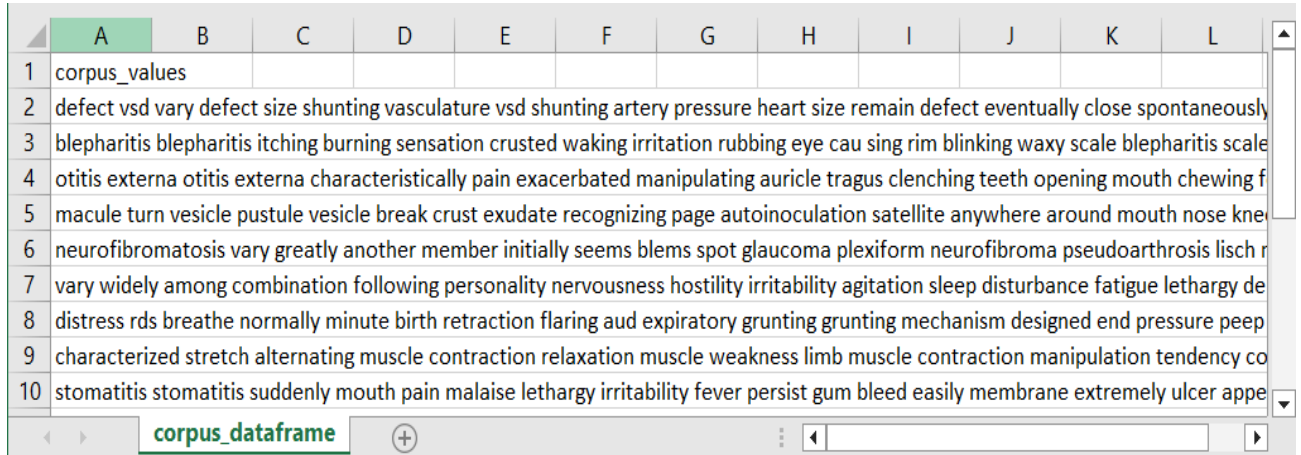
Figure 4.1 Data Source

The text is extracted and undergoes pre-processing. This includes:

- **Tokenization:** Tokenization the process of converting sentences into understandable bits of data, in this, case, words. The tokens are filtered to exclude digits and special characters and converted to lower case.
- **Stop Words Removal:** Stop words are a set of commonly used words in every language. Such words, along with stops wors specific to the medical domain are filtered out.

- **Lemmatization:** Lemmatization is the process of grouping together different inflected forms of the same word by breaking them down to their root.

The pre-processed data is stored in a CSV file as shown in figure 4.2.



	A	B	C	D	E	F	G	H	I	J	K	L
1	corpus_values											
2	defect vsd vary defect size shunting vasculature vsd shunting artery pressure heart size remain defect eventually close spontaneously											
3	blepharitis blepharitis itching burning sensation crusted waking irritation rubbing eye cau sing rim blinking waxy scale blepharitis scale											
4	otitis externa otitis externa characteristically pain exacerbated manipulating auricle tragus clenching teeth opening mouth chewing f											
5	macule turn vesicle pustule vesicle break crust exudate recognizing page autoinoculation satellite anywhere around mouth nose kne											
6	neurofibromatosis vary greatly another member initially seems blems spot glaucoma plexiform neurofibroma pseudoarthrosis lisch r											
7	vary widely among combination following personality nervousness hostility irritability agitation sleep disturbance fatigue lethargy de											
8	distress rds breathe normally minute birth retraction flaring aud expiratory grunting grunting mechanism designed end pressure peep											
9	characterized stretch alternating muscle contraction relaxation muscle weakness limb muscle contraction manipulation tendency co											
10	stomatitis stomatitis suddenly mouth pain malaise lethargy irritability fever persist gum bleed easily membrane extremely ulcer appe											

Figure 4.2 Corpus

## 4.2.2 Topic Modelling

Topic Modelling is a type of statistical modelling that uses unsupervised machine learning to identify clusters or groups of similar words within a body of text. Through this, the keywords that best describe each kind of disease are identified. The topic modelling technique used in this project is Latent Dirichlet Allocation (LDA). The results are stored in a separate CSV file as shown in figure 4.3.

	A	B	C
1	Description	D_Name	
2	pain, heart, muscle, fever, blood, pressure, dyspnea, vomiting, weakness	Cardiologist	
3	seizure, brain, headache, weakness, leg, motor, eye, speech, difficulty, co	Neurologist	
4	pain, bowel, vomiting, stool, diarrhea, reflux, obstruction, tenderness, he	Gastroenterologist	
5	cleft, skin, hair, defect, person, lip, abnormality, crisis, birth, bone	Medical Geneticist	

Figure 4.3 Doctors and Descriptions

### 4.2.3 Cosine similarity

Cosine similarity is a metric used to measure how similar two text-based documents are, irrespective of its size. The techniques used to predict the type of disease are:

- **Count Vectorizer:** It counts the number of times a word appears in a document using the bag-of-words approach.
- **TF-IDF Vectorizer:** It shows how important that word is to that chapter.

A sample output from performing cosine similarity is shown in 4.4

```
This is chapter number : 8  
Cosin cv :      0.30151134457776363  
Cosin TFIDF : 0.32836812811814714
```

Figure 4.4 Cosine CV and TF-IDF

### 4.2.4 Classification and prediction

The classification technique used here is Logistic Regression. It is a supervised machine learning algorithm mainly used for classification tasks where the goal is to predict the probability of an instance belonging to a given class.

From this analysis, the system suggests the healthcare professional that the patient will benefit from visiting the most.

## 4.4 ADVANTAGES

### (a) Availability around the clock:

Doctors are vital and that they do their best to be available all the time and dedicate enough attention to every patient. But the matter is, doctors are usually on a decent schedule, and being available for each patient is impossible sometimes. Hence, Medical Diagnosis System came into work. Medical

Diagnosis System are available 24/7 and they're personally dedicated to assisting you throughout your recovery. While the doctors save more lives out there, this Medical Diagnosis System can assist you in providing medical information.

**(b) Cost Effectiveness for Patients:**

All Medical Diagnosis System within the medical line need to be presented rightly and made as attractive as possible. When a patient visits the hospital, the Medical Diagnosis System will help them through the symptoms, predict potential diagnosis. This reduces the need to consult several doctors and pay visiting charges for each before identifying the disease.

**(c) Time Effectives for Patients:**

In case of an emergency, it's important that a patient is directly delivered to the hospital. However, if things aren't as clear then it'd consume an excessive amount of time to just identify the doctor required, particularly if the patient travels all the time to the hospital or clinic. This is often where Medical Diagnosis System save valuable time. A Medical Diagnosis System can now detect a patient's ailment by getting the user's symptoms as input to the system.

## CHAPTER 5

### ARCHITECTURE AND DESIGN

#### 5.1 SYSTEM ARCHITECTURE

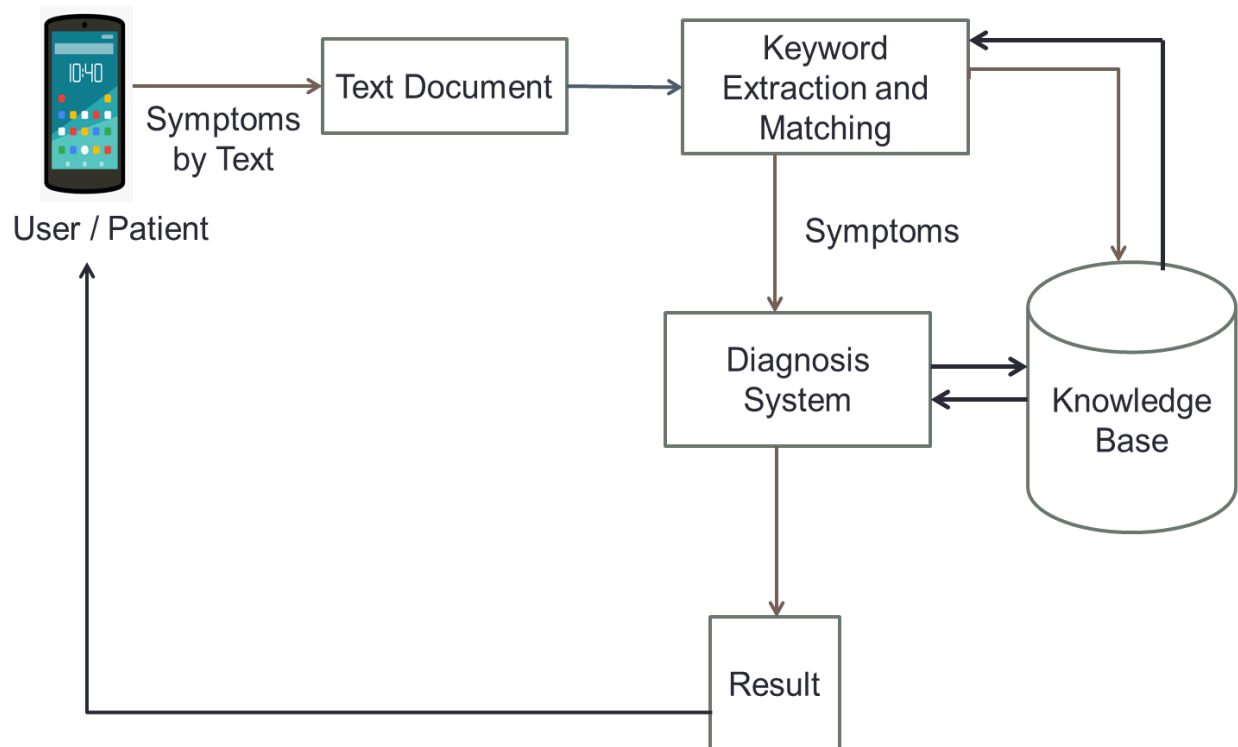


Figure 5.1 Sequential Model

The steps involved in the proposed medical diagnosis system are as follows:

1. Description of the knowledge base
2. Preprocessing of text-based documents
3. Tagging of document
4. Extraction of answer
5. Ranking of candidate answers.

- **KEYWORD EXTRACTION**

Symptoms are collected and perform preprocessing steps are to remove the noisy words. The basic steps are

- **TOKENIZATION**

The given document is considered as a string and identifies single word in the document

- **REMOVAL OF STOP WORDS**

In this step the removal of usual words like a, an, but, and, of, the, etc. is done.

- **STEMMING**

A stem is a natural group of words with equal meanings. This method describes the base of a particular word.

## 5.2 Data Flow Diagram

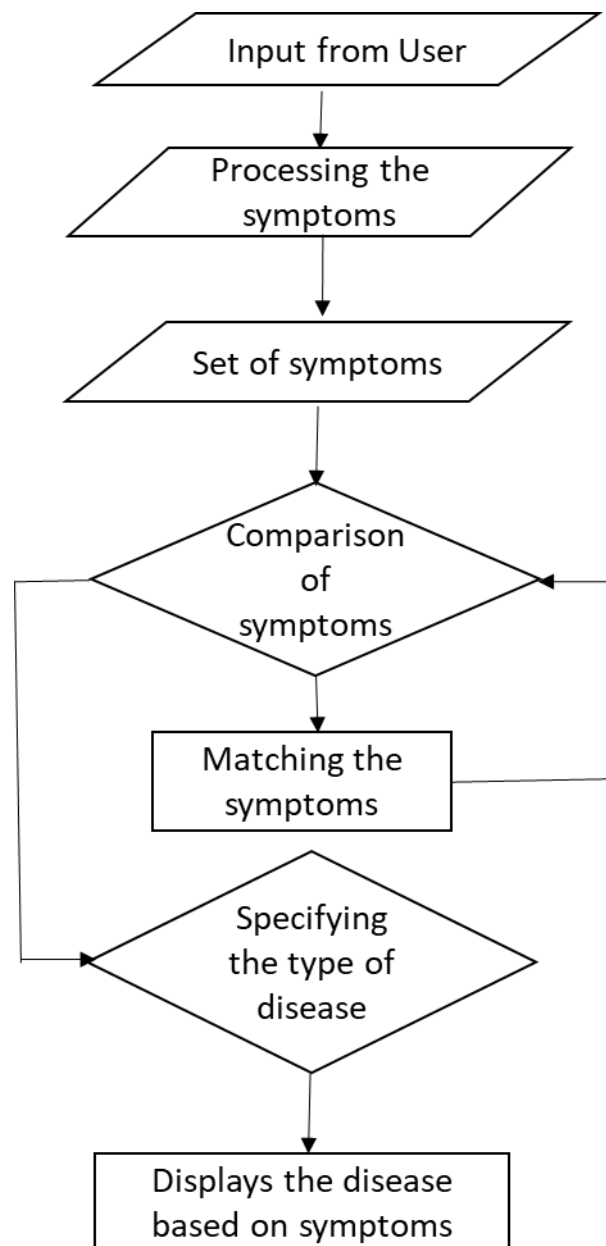


Figure 5.2 Data Flow Diagram

A data flow diagram shows the way information flows through a process or system. It includes data inputs and outputs, data stores, and the various subprocesses the data moves through.

### 5.3 Use Case Diagram :

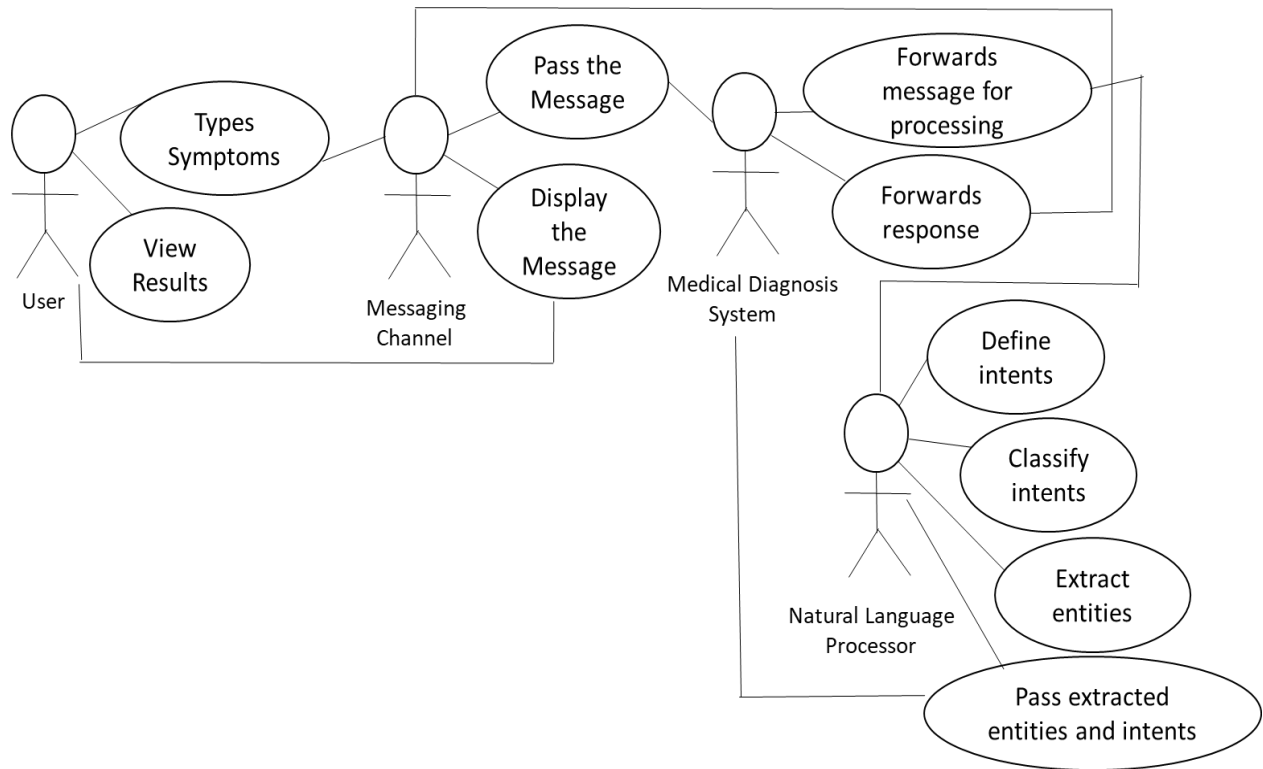


Figure 5.3 Use Case Diagram

A use case diagram is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of users the system has and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses.



## 5.4 Sequence Diagram

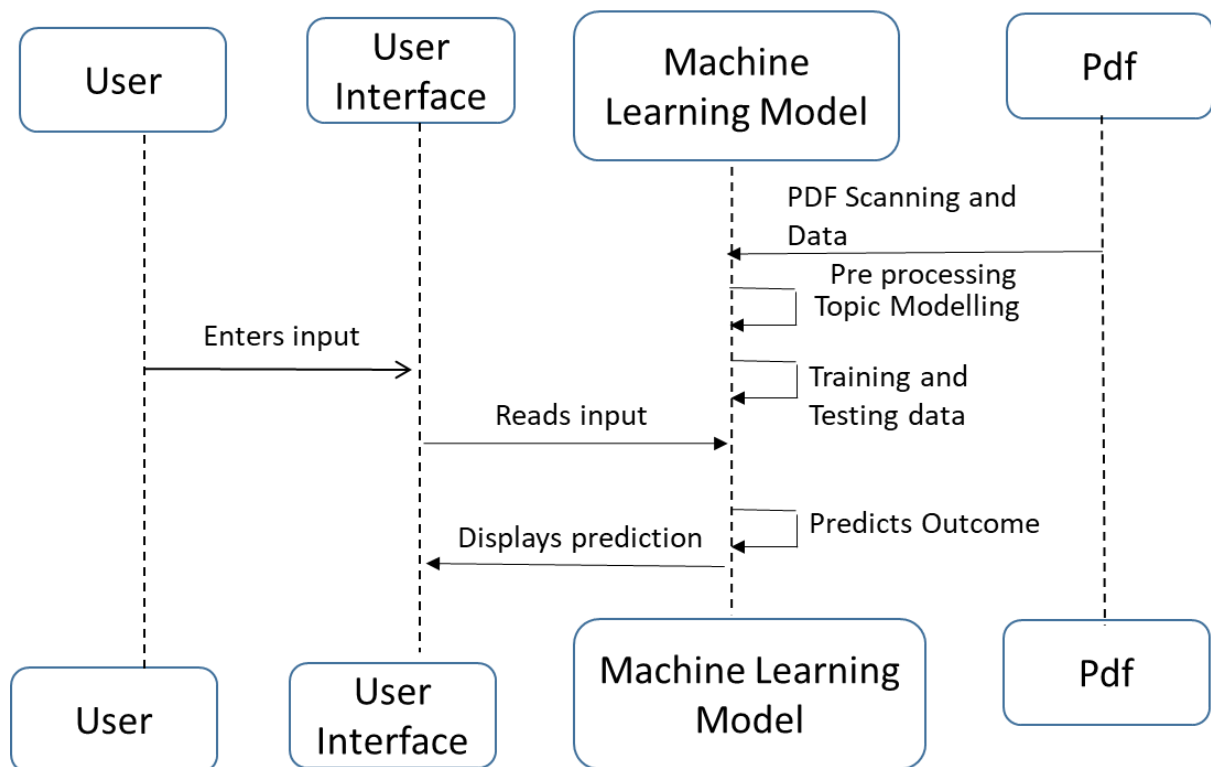


Figure 5.4 Sequence Diagram

A sequence diagram is a type of interaction diagram because it describes how—and in what order—a group of objects works together. These diagrams are used by software developers and business professionals to understand requirements for a new system or to document an existing process.

## 5.5 Class Diagram

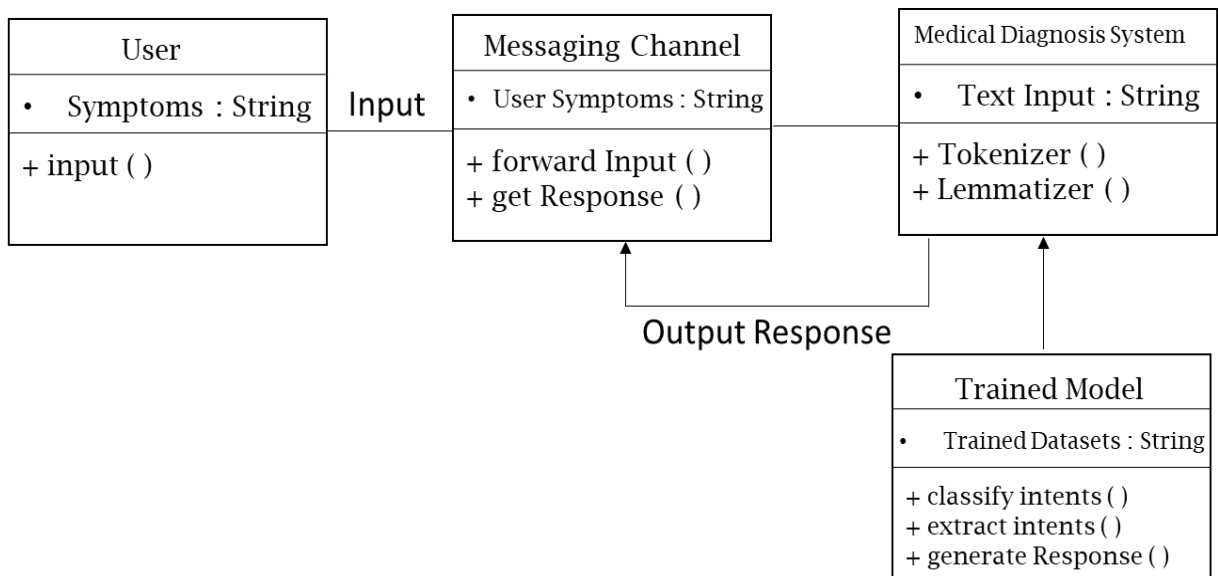


Figure 5.5 Class Diagram

The class diagram is the main building block of object-oriented modelling. It is used for general conceptual modelling of the structure of the application, and for detailed modelling, translating the models into programming code. Class diagrams can also be used for data modelling.

## 5.6 Interaction Diagram

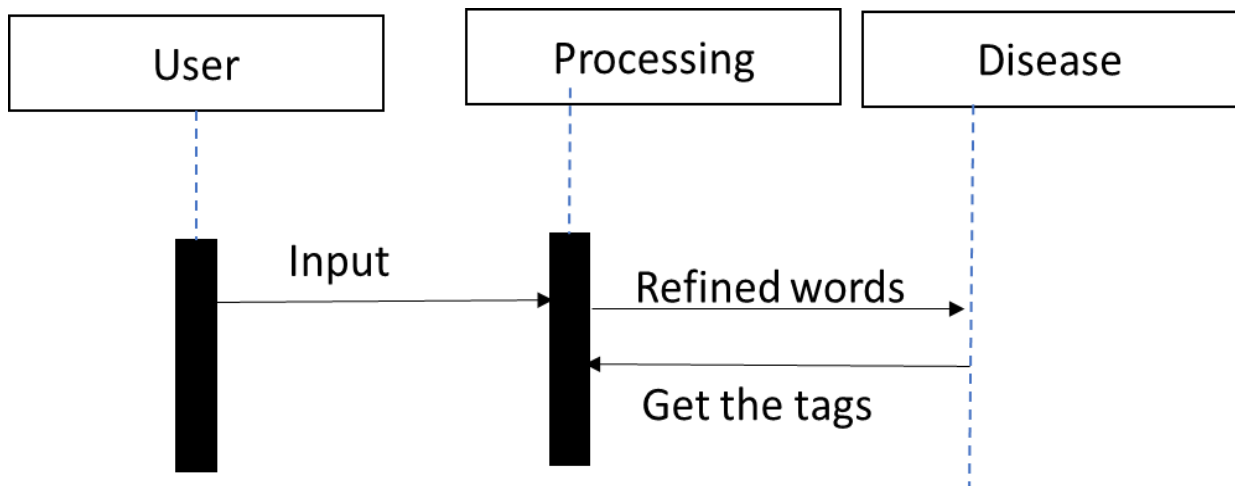


Figure 5.6 Interaction Diagram

Interaction diagrams are models that describe how a group of objects collaborate in some behaviour - typically a single use-case. The diagrams show a number of example objects and the messages that are passed between these objects within the use-case.

## 5.7 State or Activity Diagram

A state diagram is a type of diagram used in computer science and related fields to describe the behaviour of systems. State diagrams require that the system described is composed of a finite number of states; sometimes, this is indeed the case, while at other times this is a reasonable abstraction

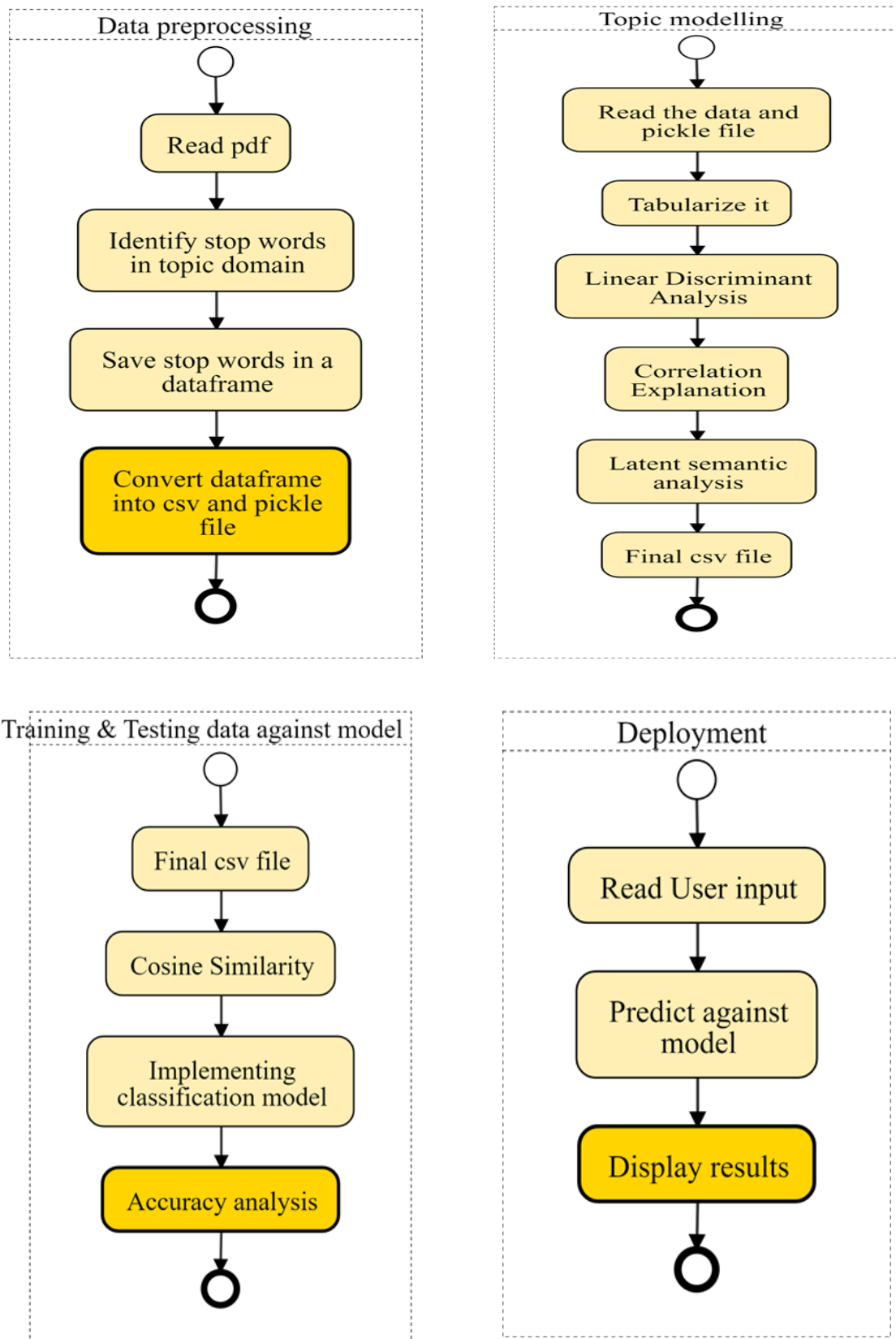


Figure 5.7 State/Activity Diagram

## 5.8 Component and Deployment Diagram

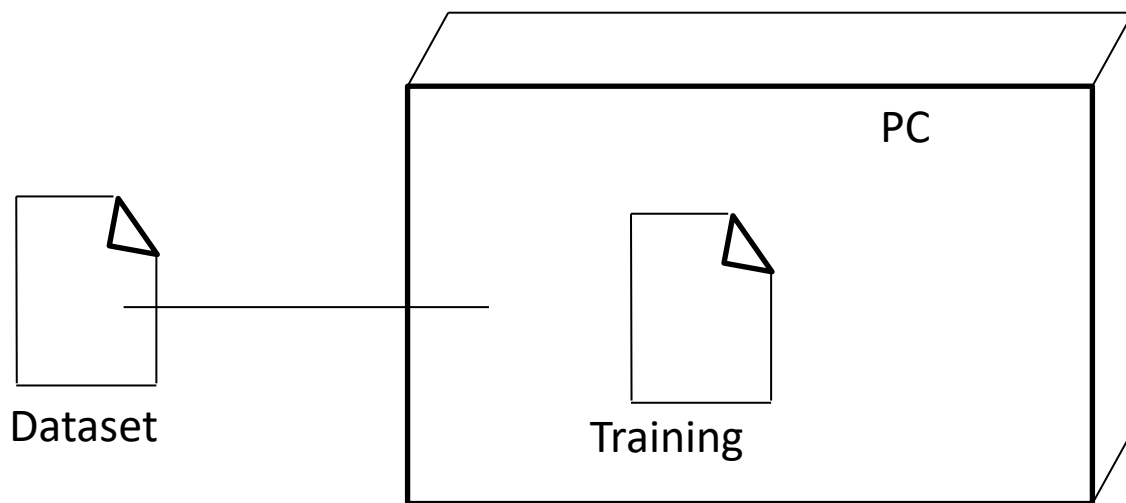


Figure 5.8 Component & Deployment Diagram

The term Deployment itself describes the purpose of the diagram. Deployment diagrams are used for describing the hardware components, where software components are deployed. Component diagrams and deployment diagrams are closely related. Component diagrams are used to describe the components and deployment diagrams shows how they are deployed in hardware.

## 5.9 ER Diagram

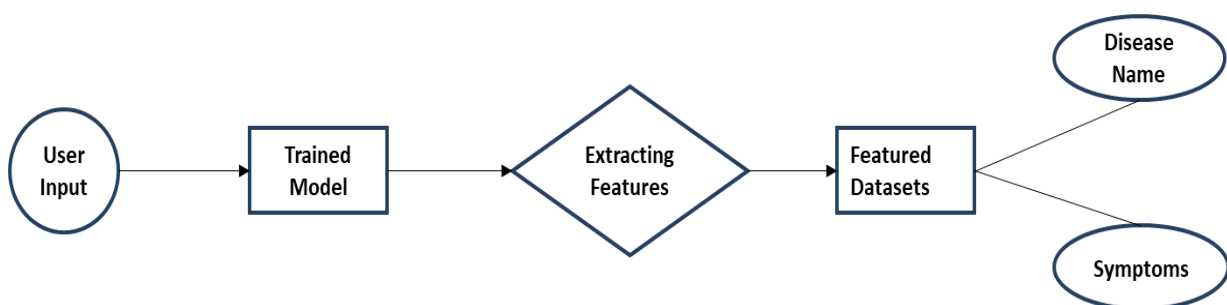


Figure 5.9 Entity-Relationship Diagram

An entity–relationship model describes interrelated things of interest in a specific domain of knowledge. A basic ER model is composed of entity types and specifies relationships that can exist between entities.

## **CHAPTER 6**

### **PERFORMANCE ANALYSIS**

#### **6.1 Unit Testing**

Unit testing involves the planning of test cases that validate that the interior program logic is functioning properly, which program inputs produce valid outputs. All decision branches and internal code flow should be completely validated. It's the testing of individual software units of the appliance .it is done after the completion of a private unit before integration. This is often a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a selected business process, application, and/or system configuration. Unit tests make sure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results. Unit testing is typically conducted as a part of a combined code and unit test phase of the software lifecycle, although it's not uncommon for coding and unit testing to be conducted as two distinct phases.

#### **Test strategy and approach**

Testing should be performed on each individual modules and should be evaluated based on error-free performance.

#### **Test objectives**

- All field entries must work properly.
- Text processing to be done in User Input
- The entry screen, messages and responses must not be delayed.
- Features to be tested
- Verify that the entries are of the right format
- Duplicate entries should be removed using lemmatizer
- Output response should be generated based on User Input

Test_ID	Test Action	Steps	Input	Expected Output	Actual Output	Pass/fail
1	Check if features are extracted properly	1. Saving the coding file 2. Executing the coding file	Input Symptoms	Extracted Features	Extracted Features	pass
2	Check if features are extracted properly	1. Saving the coding file 2. Executing the coding file	Input greetings	Extracted Features	Extracted Features	pass

Table 6.1 Sample Unit Testing

## 6.2 Integration Testing

Integration tests are designed to check integrated software components to work out if they really run together. Testing is event driven and is more concerned with the essential outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the mixture of components is correct and consistent. Integration testing is specifically aimed toward exposing the issues that arise from the mixture of components.

**Test Results:** All the test cases mentioned above passed/executed successfully. No defects has been found.

## 6.3 User Testing

User testing is the process through which the interface and various functions of a website, applications, products, or services are tested by real users who perform specific tasks in realistic conditions.



## EXPERIMENT RESULTS AND ANALYSIS

### 6.4 Result

The system was integrated with a gradio interface and provided with sample inputs as a user would provide it with the results are as follows :

INPUT	RESULT
Itchiness in scalp,excessive hairfall	<p>Type of doctor Predictor</p> <p>Enter your symptoms and get disease predictions with confidence probabilities.</p> <div> <div> Detail your symptoms:  itchiness in scalp,excessive hairfall </div> <div> <input type="button" value="Clear"/> <input type="button" value="Submit"/> </div> </div> <div> <div> <input type="button" value="Flag"/> </div> <div> <input type="button" value="Feedback"/> </div> </div> <p>Dermatologist</p>
Menstrual cramps,breast pain	<p>Type of doctor Predictor</p> <p>Enter your symptoms and get disease predictions with confidence probabilities.</p> <div> <div> Detail your symptoms:  menstrual cramps, breast pain </div> <div> <input type="button" value="Clear"/> <input type="button" value="Submit"/> </div> </div> <div> <div> <input type="button" value="Flag"/> </div> <div> <input type="button" value="Feedback"/> </div> </div> <p>Ob-Gyn</p>
Headache	<p>Type of doctor Predictor</p> <p>Enter your symptoms and get disease predictions with confidence probabilities.</p> <div> <div> Detail your symptoms:  headache </div> <div> <input type="button" value="Clear"/> <input type="button" value="Submit"/> </div> </div> <div> <div> <input type="button" value="Flag"/> </div> <div> <input type="button" value="Feedback"/> </div> </div> <p>Neurologist</p>
Yellow skin,itching,brown-coloured urine	<p>Type of doctor Predictor</p> <p>Enter your symptoms and get disease predictions with confidence probabilities.</p> <div> <div> Detail your symptoms:  Yellow skin,itching,brown-colored urine </div> <div> <input type="button" value="Clear"/> <input type="button" value="Submit"/> </div> </div> <div> <div> <input type="button" value="Flag"/> </div> <div> <input type="button" value="Feedback"/> </div> </div> <p>Medical Geneticist</p>

Table 6.4 Inputs and predicted Results

## 6.5 Result Analysis

It is clear that the model's performance isn't the best. This is due to the lack of proper data source. Due to the unavailability of properly formatted medical textbooks and implementation of less appropriate NLP techniques, the data to be trained upon is lesser and the quality is also poor. The cosine similarity, logistic regression model seems to perform well for inputs with words similar to that in the keyword extraction dataset, however even "ear ache" and "earache" seem to produce different results. It can be deduced that due to insufficient training data, overfitting occurs i.e the model memorizes the small dataset instead of learning general patterns. This may lead to biased or incomplete results as shown below :

INPUT 1	INPUT 2
<p style="text-align: center;">Type of doctor Predictor</p> <p>Enter your symptoms and get disease predictors with confidence probabilities.</p> <div> <div> <p>Detail your symptoms</p> <input type="text" value="earache, hearing loss, fever"/> <p>Clear Submit</p> </div> <div> <p>Output</p> <p>Cardiologist</p> <p>Flag</p> </div> </div>	<p style="text-align: center;">Type of doctor Predictor</p> <p>Enter your symptoms and get disease predictors with confidence probabilities.</p> <div> <div> <p>Detail your symptoms</p> <input type="text" value="earache, hearing loss, fever"/> <p>Clear Submit</p> </div> <div> <p>Output</p> <p>ENT doctor</p> <p>Flag</p> </div> </div>

Table 6.5 Predicted results comparison

It is also difficult to provide evaluation metrics for the model , since it tends to overfit.

Therefore, it is inferred that a better data source should be used and better NLP techniques can be implemented so that better results could be provided. With a larger dataset and better quality, text pre-processing, feature extraction and prediction processes' will be better hence increasing the model's quality and performance.

## **CHAPTER 7**

### **CONCLUSION AND FUTURE WORK**

#### **7.1 Conclusion**

Lots of research is going on in the field of extraction of medical text with the help of NLP. As medical text is different than normal text, it needs advanced tools as compared to the normal NLP tools. We implemented the system which can give the basic information regarding diseases and also can give the disease information onto the basis of diseases. Medical Diagnosis System is useful application for predicting the acquired disease. The project is developed for getting a fast response from the system which suggests with none delay it gives the accurate result to the user. Thus, it has wide and vast future scope.

#### **7.2 Future Scope**

The implementation of personalized medicine would successfully save many lives and make a medical awareness among the people regardless of how far people are, they will have this medical conversation. The sole requirement that they have may be a simple desktop or smartphone with internet connection. The efficiency of the diagnosis system will be often improved by adding more combination of words and increasing the utilization of database. This system could be modified to identify the particular disease instead of just the type of disease. Even though computer programs using textual mediums are growing popular among healthcare institutions, there are challenges to the use of NLP in medicine. Clinicians will need training to understand how NLP can be safely used as part of routine practice. In the future, NLP applications are likely to be integrated into the clinical environment, working with clinicians to improve effectiveness of diagnosis. With improvement, this tool could become the perfect assistant in healthcare industries.

## CHAPTER 8

### APPENDICES

#### APPENDIX 1 – SAMPLE CODE :

```
import pandas as pd
from sklearn.linear_model import LogisticRegression
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
import re
import pickle
from numpy import dot
from numpy.linalg import norm

# reading the stop words list with pickle
with open ('stop_words.ob', 'rb') as fp:
    domain_stop_word = pickle.load(fp)

# read data file
file_path = 'diseases_with_description.csv'
df = pd.read_csv(file_path)
#print(df.head())

def clean_text_func(text):

    """ this function clean & pre-process the data """

    text = str(text)
```

```

text = text.lower()
# Clean the text
text = re.sub(r"[^A-Za-z0-9^,!?.\/'\+]", " ", text)
text = re.sub(r"\+", " ", text)
text = re.sub(r",", " ", text)
text = re.sub(r"\.", " ", text)
text = re.sub(r"!", " ", text)
text = re.sub(r"\?", " ", text)
text = re.sub(r"\"", " ", text)
text = re.sub(r":", " : ", text)
text = re.sub(r"\s{2,}", " ", text)
text = re.sub(r"[0-9]", " ", text)
final_text = ""
for x in text.split():
    if x not in domain_stop_word:
        final_text = final_text + x + " "
return final_text

```

```

df['Description'] = df['Description'].apply(lambda x: clean_text_func(x))
df.head()

```

**# WORDS EMBEDDING**

```

cv = CountVectorizer(stop_words="english")
cv_tfidf = TfidfVectorizer(stop_words="english")

X = cv.fit_transform(list(df.loc[:, 'Description' ]))
X_tfidf = cv_tfidf.fit_transform(list(df.loc[:, 'Description' ]))

```

```

df_cv = pd.DataFrame(X.toarray() , columns=cv.get_feature_names_out())

df_tfidf = pd.DataFrame(X_tfidf.toarray() ,
columns=cv_tfidf.get_feature_names_out())

#print(df_cv.shape)

cosine = lambda v1 , v2 : dot(v1 , v2) / (norm(v1) * norm(v2))

# Cosine Similarity

new_text = [input('Detail your symptoms:\n')]
new_text_cv = cv.transform(new_text).toarray()[0]
new_text_tfidf = cv_tfidf.transform(new_text).toarray()[0]

for chapter_number in range(int(df.shape[0])):
    print(f"This is chapter number : {chapter_number} ")
    print(f"Cosin cv : { cosine( df_cv.iloc[chapter_number] , new_text_cv )} ")
    print(f"Cosin TFIDF : { cosine( df_tfidf.iloc[chapter_number] , new_text_tfidf)
} ")

# Implementing Logical Regression

#print(df.columns)

X_train = df.Description
y_train = df.D_Name

cv1 = CountVectorizer()

```

```
X_train_cv1 = cv1.fit_transform(X_train)
pd_cv1 = pd.DataFrame(X_train_cv1.toarray(),
columns=cv1.get_feature_names_out())
```

```
lr = LogisticRegression()
lr.fit(X_train_cv1, y_train)
```

```
X_test = new_text
cleaned_text = clean_text_func(X_test)
```

```
X_test_cv3 = cv1.transform([cleaned_text])
y_pred_cv3 = lr.predict(X_test_cv3)
print(y_pred_cv3)
```

## APPENDIX 2 - SCREENSHOTS

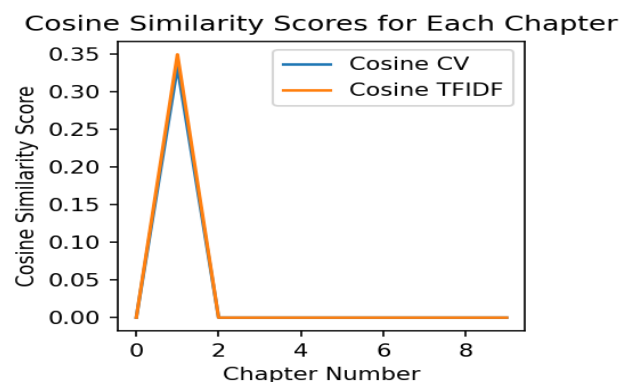
### IN IDLE SHELL:

#### 1)INPUT

```
*IDLE Shell 3.10.11*
File Edit Shell Debug Options Window Help
Python 3.10.11 (tags/v3.10.11:7d4cc5a, Apr 5 2023, 00:38:17) [MSC v.1929 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\keert\Downloads\signs_and_symptoms\final.py =====
Detail your symptoms:
Numbness in face, loss of motor control, weak movement
```

#### 2)COSINE SIMILARITY , TF-IDF

```
IDLE Shell 3.10.10
File Edit Shell Debug Options Window Help
...
>>>
===== RESTART: D:\Projects\Medical-Diagnosis-NLP-main\textbook_source\signs_and_symptoms\final.py =====
Detail your symptoms:
Numbness in face, loss of motor control, weak movement
This is chapter number : 0
Cosin cv : 0.0
Cosin TFIDF : 0.0
This is chapter number : 1
Cosin cv : 0.3333333333333333
Cosin TFIDF : 0.354305979085421
This is chapter number : 2
Cosin cv : 0.0
Cosin TFIDF : 0.0
This is chapter number : 3
Cosin cv : 0.0
Cosin TFIDF : 0.0
This is chapter number : 4
Cosin cv : 0.0
Cosin TFIDF : 0.0
This is chapter number : 5
Cosin cv : 0.0
Cosin TFIDF : 0.0
This is chapter number : 6
Cosin cv : 0.0
Cosin TFIDF : 0.0
This is chapter number : 7
Cosin cv : 0.0
Cosin TFIDF : 0.0
This is chapter number : 8
Cosin cv : 0.0
Cosin TFIDF : 0.0
['Neurologic']
```





## IN USER INTERFACE:

### Type of doctor Predictor

Enter your symptoms and get disease predictions with confidence probabilities.

Detail your symptoms:

Numbness in face, loss of motor control, weak movement

Clear

Submit

output

Neurologist

Flag

## REFERENCES

- 1 Kyle D. Richardson<sup>1</sup>, Daniel G. Bobrow<sup>1</sup>, Cleo Condoravdi<sup>1</sup>, Richard Waldinger<sup>2</sup>, Amar Das<sup>3</sup>, “English Access to Structured Data”, 2011 Fifth IEEE International Conference on Semantic Computing
- 2 Faguo ZHOU Enshen WU, “The Design of Computer Aided Medical Diagnosis System Based on Maximum Entropy” 978-1-61284-729-0111 2011 IEEE
- 3 Stéphane Meystre, Peter J. Haug, “Natural language processing to extract medical problems from electronic clinical documents: Performance evaluation”, Journal of Biomedical Informatics 39 (2006) 589–599
- 4 Stéphane Meystre, Peter J Haug, R. Engelbrecht et al., “Evaluation of Medical Problem Extraction from Electronic Clinical Documents Using MetaMap Transfer (MMTx) Connecting Medical Informatics and Bio-Informatics”, ENMI, 2005
- 5 Lucila Ohno-Machado, Editor-in-chief, Prakash Nadkarni, Kevin Johnson “Natural language processing: algorithms and tools to extract computable information from EHRs and from the biomedical literature”, amiajnl-2013-002214
- 6 Khan Razik, Dhande Mayur , “To Identify Disease Treatment Relationship in Short Text Using Machine Learning & Natural Language Processing”, Journal of Engineering, Computers & Applied Sciences (JEC&AS), Volume 2, No.4, April 2013
- 7 Wafaa Tawfik Abdel-moneim<sup>1</sup>, Mohamed Hashem , “Clinical Relationships Extraction Techniques from Patient Narratives”, JCSI

- 8 Asma Ben Abacha, Pierre Zweigenbaum, “Automatic extraction of semantic relations between medical entities: a rule based approach” From Fourth International Symposium on Semantic Mining in Biomedicine (SMBM) Hinxton, UK. 25-26 October 2010
- 9 Stéphane M. Meystre, MD, MS, Peter J. Haug , “Comparing Natural Language Processing Tools to Extract Medical Problems from Narrative Text”, MD AMIA 2005 Symposium Proceedings
- 10 Jiaping Zheng,<sup>1</sup> Wendy W Chapman,<sup>2</sup> Timothy A Miller,<sup>1</sup> Chen Lin, “A system for coreference resolution for the clinical narrative”, J Am Med Inform Assoc (2012). doi:10.1136/amiajnl-2011- 000599
- 11 Romer Rosales, Faisal Farooq, Balaji Krishnapuram, Shipeng Yu, Glenn Fung, “Automated Identification of Medical Concepts and Assertions in Medical Text Knowledge Solutions” , AMIA i2b2/VA text mining challenge
- 12 D . Nagarani, Avadhanula Karthik, G. Ravi, “A Machine Learning Approach for Classifying Medical Sentences into Different Classes”, IOSR Journal of Computer Engineering (IOSRJCE) Volume 7, Issue 5 (Nov-Dec. 2012), PP 19-24
- 13 L. Smith<sup>1</sup>, T. Rindflesch<sup>2</sup> and W. J. Wilbur, “MedPost: a part-of-speech tagger for bioMedical text”, Vol. 20 no. 14 2004, pages 2320–2321, bioinformatics/bth227
- 14 Dan Shen Jie Zhang Guodong Zhou, “Effective Adaptation of a Hidden Markov Model-based Named Entity Recognizer for Biomedical Domain
- 15 Nicholas A. I. Omoregbe, Israel O. Ndaman, Sanjay Misra, Olusola O. Abayomi-Alli, Robertas Damaševičius, "Text Messaging-Based Medical Diagnosis Using Natural Language Processing and Fuzzy

Logic", *Journal of Healthcare Engineering*, vol. 2020, Article ID 8839524, 14 pages, 2020. <https://doi.org/10.1155/2020/8839524>

- 16 R. B. Mathew, S. Varghese, S. E. Joy and S. S. Alex, "Chatbot for Disease Prediction and Treatment Recommendation using Machine Learning," 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 2019, pp. 851-856, doi: 10.1109/ICOEI.2019.8862707.