Report for MS Thesis

Formulating Patient Descriptions to Support Evidence Based Medicine (EBM)

By

Muhammad Suffian

FA16-MSCS-0043

fa16mscs0043@maju.edu.pk

Supervisor

Dr. Shaukat Wasi

Head of Department

Computer Science

Mohammad Ali Jinnah University, Karachi

[20th May 2018]

Table of Contents

List of figures

List of tables

ABSTRACT

The evidence based medicine is the helpful practice for medical practitioners to take decisions with available evidences and their expertise. After identification of the problems and ambiguities faced by practitioners, the natural language processing solutions are proposed by the experts to improve EBM. One of the case is the patient medical records or descriptions in textual format to process and to extract semantics, then to formulate the queries for best evidence based medicine practice. Our system will utilize the natural language processing and the suite of machine learning algorithms to extract the meaningful information from the patient descriptions like disease class and then formulate the query to propose the available medicine.

Chapter 1

1- INTRODUCTION

The idea of Evidence Based Medicine (EBM) caused incredible enthusiasm among wellbeing experts. As indicated by definition Evidence Based Medicine is the medication suggested by the doctors underlying the available health status of patient by formulating the question or query accordingly and then applying on the corpus of medical journals to retrieve the summaries or results related to the disease[1]. The reason for consulting the medical journals is because the medical practitioners have to get aligned with the day by day new achievements published in medical journals. The current technological advancements have revolutionized the EBM concept. This helped the doctors to opt the latest curing medications for the severe type of diseases. In spite of many hurdles, Evidence Based Medicine practice has gained reputation over recent years due to the reasons, like the improvements in patient's healthcare. Research advancements are removing the barriers in EBM and it is inferred that the boom will come with NLP techniques.

Problem Statement / Motivation

Our problem is an inspiration from Sarker et al's work [2]. They discussed the problems and obstacles in evidence based medicine faced by the practitioners. They categorized the problems in five major parts. One of those problems is related to formulate the question or query that should include all important information without ambiguity and about the information retrieval. The Névéol et al [25] had identified the opportunities and challenges to work with clinical natural language processing. They had also described the problems with different methods/algorithms with respect to language context.

Natural language processing can do helpful things for the evidence based medicine. The current research in medical information retrieval has concentrated on query design and other facets of

information retrieval to support practitioners. The sentences in form of patient descriptions spoken or written by the patient are very important for the doctor and the machine/robot to instruct/suggest/search the medication strategy from the large medical corpus or using the own skill set based on experience. The very first thing to help doctors/machines to formulate the query/strategy needs the semantic extraction or information extraction from the sentences uttered/written by the patient. Here involves the natural language processing. The correct or true information searched or retrieved by the doctor/machine depends on the correctness of the formulation of query or the understanding developed by the doctor/machine from the sentence. The first reason is that most of the doctors and machines/robots cannot formulate the correct query because of the ambiguity in sentences due to the multiple meanings of the sentence. Second possible reason can be the less awareness of technology to doctors i.e. how to search or retrieve the information results from the corpus? Now this problem of query formulation can be fixed using the natural language processing and machine learning techniques and in this way the precision and recall of searched query can be increased.

Chapter 2

2- LITERATURE REVIEW

An approach similar in spirit to our work is discussed and modeled QRAQ [3] (Guo, Xiaoxiao, et al 2017) in which the user story as text and the challenging question is given to the agent that deduce the information from the text with existing ambiguities, and it should be able to answer the question. If the agent cannot answer then firstly it learns and deduce the variables from the fact in the problem. Secondly, if agent cannot answer the question by reasoning alone then it infers from the simulator to extract the other variables from the problem and should be relevant to question. The problem domain of this work is similar to our domain work. They used the Reinforcement Learning (RL) approach in their work and based on (RL) they presented and evaluated two memory network architectures. Our work is more towards Natural language processing machine learning.

In [4] (Molla et al 2016) built a corpus for the text processing. They have taken the data set from the clinical inquiries segment of the journal dealing family practice [5]. They annotate the data using the annotation techniques like automatic extraction, manual annotation and the rephrasing text. The inquiry sentence is used as query and the retrieval text then summarized to answer. The summary of text is basically is divided into few sentence classes and the human annotation was used to classify them into according summary. They associated three evidence based answers to each question and each answer deal with separate evidence. The criteria of suggestion is based on the score of matching to the evidence. In the work of Molla et al [4] one thing can increase the accuracy of the retrieved summaries that is the removal of ambiguity from the input sentence/query.

In [6] (İlknur DÖNMEZ et al) formed a phrase-content finder system for the Turkish sentences. They have done this study by underlying the importance of subject, verb and object relation with actionable things. The phrase content relationship is also valuable because of its structural importance for sentence. They divided the sentence mainly into two parts, one the phrase and the other as content. In each sentence they separated it into 8 different phrases, then if the phrase exists the concepts are determined from the database like Word-Net. These phrase-concepts pairs like syntactic and semantic information of sentences have shown with matrix representation.

A Question/Answer system [7] (Avani et al 2017) is built focusing on the structured and annotated knowledge base. The system is divided into three parts question processing, information retrieval and the answer extraction. The question processing part is related to my study that is divided into two parts: First the question is given to python factoid question classifier [8] this determines the type of the question and also the category of answer to this question, Second the question is parsed using the Stanford dependency parser which checks the dependencies of words and POS tagging is done in parallel. In this way they determined the focus of the question. But they also highlighted the limitation of this approach that python factoid classifier does not categorized the questions in which there is a call for action. They evaluated their Question/Answer system on TREC 2004 question data set.

In [8] (kim et al 2011) build a sentence classifier that firstly identifies the key sentences and then classifies them with medical tags. Their classifier uses conditional random fields CRFs for the learning algorithm purposes. The classifier is trained with basically four features lexical information, semantic information, structural information and sequential information. In lexical information feature they used the bag of words with bigrams and then applied POS tagging for the semantic similarity in two texts. In semantic information the metathesaurus from UMLS was used,

then directly query the thesaurus with each input token. MetaMap analyzer used for sentence parsing, in this way they get the concept unique identifiers and identified the same text. The corpus was 1000 abstracts and each sentence was annotated. I highlighted only the relevant work of kim et al, their work is more towards the sentence classification retrieved from the abstracts. The features like lexical and semantic information are more related, but utilized on results after querying, the ambiguity of query and question meaning before applying on data set is not handled in their work.

Abeed Sarker et. al. presented a query focused approach for text summarization to support evidence based medicine [9]. The query specific summaries were extracted by introducing a scoring scheme in which the score was assigned to sentence on UMLS type and the category type it contains. Semantic type information improved the extractive summarization performance. They classified the questions in their corpus into medical topics using the approach [10]. For the better question associations with summaries they sets two semantic types for each question (a) important question semantic types that were identified during training and (b) important answer semantic types that is identified from human authored summaries in training. They evaluated their approach using ROUGE evaluation tool, their QSpec system outperforms previous systems working on same perspective with 96.5 % percentile rank. But the (Sarket et al) also highlighted the room for improvement that can be achieved by improving intermediate steps for the feature generation in summarization task.

In [11](pratt et al 2000) a new approach for categorizing the search results was implemented with the name DynaCat system. In this they divided the semantics of dynamic categorization into two models (a) small query model that keeps the knowledge of the types of queries users make (b) a large domain specific terminology model, Dynacat uses UMLS for handling large terms and their

synonyms. In query model the algorithm takes the types of queries and check the category of relevant query types. The limitation of query model is, it independent of disease specific terms means it generalizes the query into the specific category like categorizing in 'treatment type' or 'adverse effect' etc. This system was made for the patients and their family members with a questionnaire form to input the query data. This system was claimed better than previous ranking based and clustering based models. In this work the query or question from the patient was taken but the processing on it is not more to clear the sentence level ambiguities and to did not assigned the category on the basis of disease.

In [12] (Cao, Yonggang, et al 2011) developed an online system that is related to question answering in a complex clinical query environment, AskHERMES is a system that is in comparison with google and upToDate system for complex questions to answer with beating accuracy. Their complex question handling part is the NLP and IR problem and they have handled it with UMLS and CRFs. The system worked on vast datasets like Medline, PubMed, eMedicine etc. This system limit is highlighted by the Cao et al that is it does not integrate the complex clinical evidence identification part that is entered by upToDate manually.

Chapter 3

3- METHODOLOGY

The methodology for solving the problem is splitted in two halves. The different approaches used for the semantic extraction and to formulate the patient descriptions to support evidence based medicine are:

1- Language Modeling techniques (n-Grams)
2- Keyword based NLP and Machine learning approach

In figure 1.4 the block diagram depicts the use of different approaches, the upper part of diagram shows the n–Grams approach and the lower part shows the keyword extraction approach.

3.1 Language Modeling

A model that assigns the probabilities to words or sequences of words is called language model. The simplest model can be termed as *n-Gram model*. The uni-gram is each individual word having probability of occurrence, bi-gram is sequence of two words with some probability, tri-gram is sequence of three words with probability and the n-gram is sequence of n words [13].

The simple uni-grams on the plain text of a specific disease descriptions (in the form of sentences) is calculated for checking the sentence probability. The count of each word within the sentence calculated and the probability is distributed on the base of counts. The probability of the sentence is the product of each word's probability.

$$S = (w_1, w_2, w_3 ... w_i ... w_n) \quad (1)$$

The probability of sentence will be like e.i

$$P(s_{w_i}) = p(w_1 * w_2 * w_i * w_n) \quad (2)$$

In uni-grams the disadvantage is lack of context information and there is no history for co-occurrence of words or grams. According to *Markov* assumptions the history of predecessors can be stored that leads to bi-grams, tri-grams and higher grams.

$$\text{bi-gram} \sim P(w_n \mid w_{n-1}) \quad (3)$$

The bi-gram model is the probability of occurrence of a word with one word from the history. The calculation is performed by the checking the occurrence of each word with next and previous word. In this way the sentence length matters and the sequence of words also matters. If the length(vocabulary) of the sentence is n then bi-gram will lead to n^2 and the tri-gram will lead to n^3. The overall problem with the n-grams model is that if a token during the testing

phase is occurred and the model didn't seen that token during training and ultimately will assign '0' probability to this token. It effects badly the sentence probability because in the case of sentence generation or checking the sentence probability, probabilities of all uni-grams/bi-grams(higher grams) are multiplied in linear fashion. So ultimately zero probability token will lead to zero probability of the sentence. To avoid this problem different smoothing techniques can be employed like add-1 smoothing discussed in section 3.1.2.

3.1.1 Evaluation of Language Modeling on patient descriptions

Evaluation of language models is done by calculating the perplexity, lower perplexity can predict the better test data by model. In case of patient descriptions (in form of the sentences) does not performed well because the new token occurring in the test set made it less accurate. An other perspective of perplexity is the branching factor, in which the future words of each word follow the given word.

$$\mathrm{PP(S)} = P(w_1 w_2 w_3 ... w_n)^{-\frac{1}{N}} \sim \sqrt{\frac{1}{P(w_1 w_2 w_3 ... w_n)}} \qquad (4)$$

If we compute the perplexity with the help of bi-gram then the equation (4) becomes

$$\mathrm{PP(W)} = \sqrt[N]{\prod_{i=1}^{N} \frac{1}{P(w_i \mid w_{i-1})}} \qquad (5)$$

In case of patient descriptions the subject is to extract the meaning of sentence in a way that can help to make better suggestion or can help to formulate better query on the database. Also the intuition was to formulate a question by the help of n-grams for applying on the database to extract the relevant results. The n-grams didn't worked for the query creation that can later be classified into some specific disease.

3.1.2 Improvements through the Smoothing

The smoothing of probabilities can help in better prediction and the overall perplexity of sentence can improve. The different smoothing techniques for evaluation has been discussed by Zhai & John [14]. The Maximum likelihood method do not assign probability to the unseen word in document. The other smoothing technique is 'Jelinek-Mercer method' it is the fixed co-efficient interpolation that is used with maximum likelihood and collection model, some type of mixture model that linearly interpolated and shown in equation (6).

$$\mathrm{P}_{\mathrm{interp}}(w_i \mid w_{i-n+1}^{i-1}) = \lambda_{w_{i-n+1}^{i-1}} P_{ML}(w_i \mid w_{i-n+1}^{i-1}) + (1-\lambda_{w_{i-n+1}^{i-1}}) P_{\mathrm{int}erp}(w_i \mid w_{i-n+2}^{i-1}) \qquad (6)$$

In Laplace method the absolute discounting is preferred in the case when the unseen word in document is present that can be handled as unigram.

$$\mathrm{P}_{\mathrm{Absolutediscount}}(w_i \mid w_{i-1}) = \frac{c(w_{i-1}, w_i) - d}{c(w_{i-1})} + \lambda(w_{i-1}) P(w) \qquad (7)$$

The conduct of each smoothing strategy demonstrates that, in general, the execution of longer inquiries is considerably more delicate to the decision of the smoothing parameters than that of tagged sentences.

The n-grams have been used by the M Conway et al [17] by adjoining with the other semantic features to classify the disease outbreak reports. They have shown that the combination of n-grams, bag-of-words and the semantic features can increase the accuracy of classification. n-grams solely cannot help to formulate a query or to extract the meaning from the patient descriptions/sentences. Therefore an alternative approach (section 3.2) is implemented that will help to initiate the query formulation or in the meaning extraction process.

3.2 Keyword based approach

3.2.1 Keyword Extraction

The keywords are the solid form of content representation in any document. Keywords help to identify the basic theme of any document. The formulation of queries in NLP and IR finds the application of keywords. Hulth have done the work of keywords extraction from the Google webpages [15].

In our case the improvement in n-gram language modeling can work for the better classification or better perplexity of sentence/descriptions of diseases. But the query formulation or the prediction of the disease from the patient descriptions is needing the terms or keywords from the description. So we used the bi and tri-gram with keywords for the better performance of query formulation and then to diagnose the disease suggesting medicine. Stuart Rose, Dave Engel, Nick Cramer and Wendy Cowley wrote a chapter of text Mining book [16] in which they describe and compare the RAKE keyword extraction algorithm with different NLP based methodologies and algorithms with their specific use.

3.2.2 Keyword extraction techniques

The list of keyword extraction tools :

- TextRank (Graph based)
- POS with supervised ML techniques
- Chi-square measured for co-occurrences
- Rapid automatic keyword extraction (RAKE)
- Conditional random fields (CRM)

We have selected the RAKE algorithm for keyword extraction on the basis of its performance.

3.2.3 Motivation for RAKE algorithm

The motivation for RAKE algorithm is that it is highly efficient, can operate on individual and multiple types of documents, can operate on documents which do not follow grammar rules. RAKE algorithm performs comparatively better than other enlisted algorithms/strategies in section (3.2.2) when to deal with large documents. Its extraction time is less and bug free. It is also better choice in the case when the keywords are multiple of words. It provides language and context independency. The words as only feature of text in the document do not play role in classification, this critique was not new but Lewis tried in his PhD thesis studies to reduce the impact of this concept by introducing the noun phrases for classification [24]. We used keywords of patient descriptions for further classification and medication to support evidence based medicine.

3.2.4 Working of RAKE algorithm

The RAKE algorithm works on the basis of stop words, punctuations and word (phrase) delimiters. The document then parsed into candidate keywords on the basis of stop word list and delimiters.

3.2.4.1 Candidate keywords

The candidate keyword extraction is started by splitting the whole document into the parsed set of words in the form of array on the basis of delimiters and stop words, then the splitted array is parsed into the pairs of contiguous words which hold specific positions inside the document. These words known as candidate keywords.

3.2.4.2 Keyword scores

The keyword score is calculated with the help of co-occurrence graph generated for the words array. Each candidate word is placed against all the words and the co-occurrence is counted in the graph. The keyword score based on the two things mainly 1) the degree of word 2) the frequency of word. Keyword score is the ration of degree of word and the frequency of word. Now the degree of word is the co-occurrence count of candidate word with other words. The keyword score is the sum of that candidate members individual scores. If the keyword is not consists of multiple words then its ratio of degree(w) to frequency(w) is the final score. The co-occurrence graph is shown in Figure 1.1.

	Feeling	Tired	Low	Grade	Fever
Feeling	1				
Tired		1			
Low			1	1	
Grade			1	1	1
Fever				1	1

Figure 1.1: Co-occurrence graph from a patient description

The calculation of keyword score is shown Figure 1.2

	Feeling	Tired	Low	Grade	Fever
Degree of word deg(w)	1	1	2	3	2
Frequency of word freq(w)	1	1	1	1	1
Ratio of deg(w)/freq(w)	1	1	2	3	2

Figure 1.2: Word scores calculated from co-occurrence graph

The keyword score is the sum of word members, thus the keyword scores are:

Patient Description : *" I am feeling very tired and having low grade fever. "*

Keywords with rank: [('low grade fever', 9.0), ('feeling', 1.0), ('tired', 1.0)]

Extracted keywords

The keyword scores are the main role playing part to extract the keywords, so after scoring the words and finding the ratio of degree to frequency the keywords can be announced.

For example, In our case we have applied the whole process of keyword extraction on few handmade patient descriptions and then the keywords has been announced. We used 7 patient descriptions consisting of 187 total words. The keywords extracted are shown below in figure 1.3:

Extracted keywords:

1 = [('low grade fever', 9.0), ('feeling', 1.0), ('tired', 1.0)]
2 = [('stomach pain', 4.0), ('feeling hungry', 4.0), ('days', 1.0)]
3 = [('dull aching pain', 9.0), ('upper abdominal quadrant', 9.0), ('clay-colored stool presented', 9.0), ('spontaneously relieved', 4.0), ('noticed', 1.0), ('days', 1.0), ('fever', 1.0), ('jaundice', 1.0), ('passing', 1.0)]
4 = [('observed dark urine', 9.0), ('vomiting', 1.0), ('yesterday', 1.0), ('10 days', 1.0), ('age', 1.0), ('35', 0)]
5 = [('drinking tasted strange', 9.0), ('stomach flu', 3.5), ('stomach', 1.5), ('noticed', 1.0), ('beer', 1.0), ('time', 1.0), ('began', 1.0), ('feel', 1.0), ('pressing', 1.0), ('gradually', 1.0), ('lose', 1.0), ('appetite', 1.0), ('thought', 1.0), ('uh-', 1.0), ('coming', 1.0), ('weeks', 1.0)]
6 = [('breakfast', 1.0), ('suddenly', 1.0), ('fainted', 1.0), ('weeks', 1.0), ('rash', 1.0), ('body', 1.0), ('eyes', 1.0), ('skin', 1.0), ('yellow', 1.0)]
7 = [('began experiencing nausea', 9.0), ('abdominal pain', 4.0), ('darned tired', 4.0), ('double shifts', 4.0), ('young child', 4.0), ('age 26', 1.0), ('fatigue', 1.0), ('wondering', 1.0), ('working', 1.0), ('week', 1.0)]

Figure 1.3: Extracted keywords from sample patient descriptions

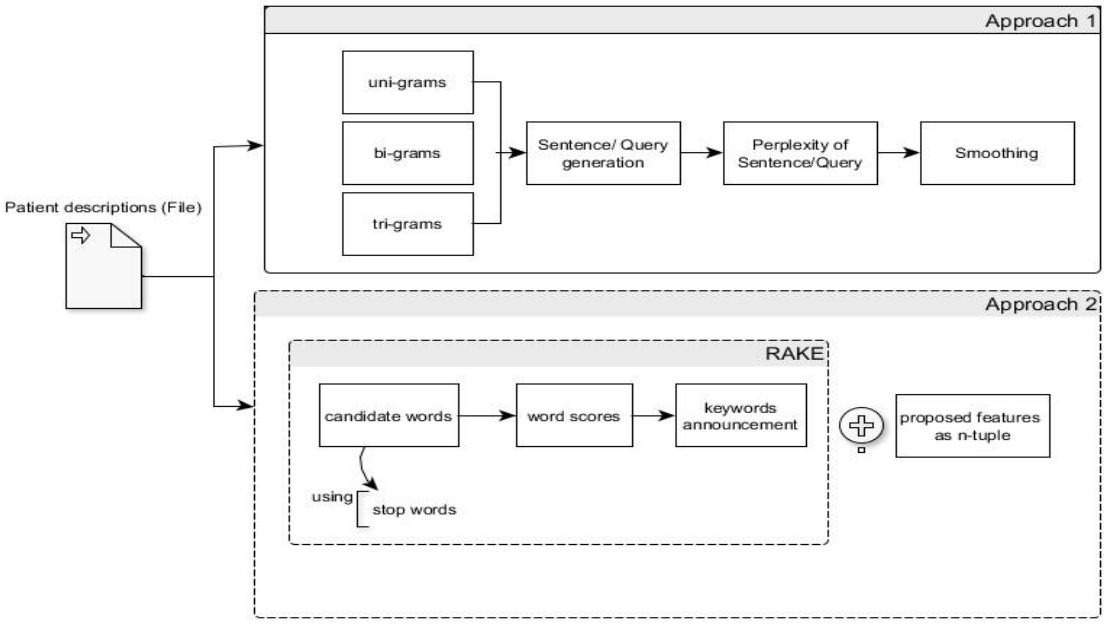

Figure 1.4: Block diagram for used approaches

The n-Grams approach was not successful for building the queries and classification of patient descriptions. The keyword approach utilized for the both tasks of patient descriptions classification and for the query formulation to suggest the medicine. The process of implementation is described in section 3.3 & section 3.4.

3.3 Classification of Patient Descriptions into Diseases

In classification the documents or text is automatically labeled with already defined labels to unseen text/documents. The text classification is used for spam filtering, e-mail filtering, document indexing and the web browsing. The text classification is very hot area of research in the world of huge unannotated text. The merger of machine learning and information retrieval has boost the classification of text. The classification of text based patient descriptions is the necessary step to progress further. The patient descriptions means the history or the records of patient in textual format need to be classified , this will also help to medical practitioners for easily diagnosing the disease first and then to take cautionary measures for further after seeing the test reports. The classification is also necessary to help in the query formulation. The query will containing the information that is related to the identified disease and also this will lead to specifically applying the query on the related target. As the description will fall in any disease category then it will be helpful to make a disease specific query. Then the query will become the target oriented. The different approaches for text classification on the basis of text features have been described in the literature. The different classification approaches underlying the problem were used. Fatima et al [18] had done survey on machine learning algorithm for diagnosing diseases where they had identified the performance of different classification algorithms and claimed the best performance of SVM in different perspectives. The RIPPER [19] was claimed to outperform the Naïve Bayes, C4.5 and K-nearest neighbors algorithms in classification on standard corpora, also they identified that the new algorithm like Support vector machines can enhance these results. So the SVM is preferable choice to work on the text classification. The diagnosing is the first step in medical checkups where the patient previous history utilized to analyze the situation the doctor prescribe something like tests or any medicine. A work is done by the kumari and chitra [20] in which they

have used the support vector machine for diagnosing the diabetes disease, they have used the PIMA dataset from UCI machine learning and got good results by applying the SVM. Chaurasia et al [21] has worked on data mining techniques for heart disease, they have used the 11 features for training. The Naïve Bayes, SVM, Decision Trees and Bagging techniques were used as classifiers. They used the UCI data set of heart disease for their research work and the processing tool was used Weka. From their acclaimed results the bagging technique performed well as compared to other classification techniques.

3.3.1 SVM as Text Classifier for patient descriptions

There are multiple text classifiers available for text classification. This should be clear that which classifier will be the main focused in our study, for explaining this we have opted the SVM for text classification/categorization. In many situations the performance of SVM outperforms but in few cases it works less. The patient descriptions are in the textual format so to work on these text based descriptions the features are extracted that will be the input to SVM. Another reason from the reasons of motivation is that SVM handles the features very smoothly whether they are large or small. It victories the features in the space in different ways on the basis of its kernel functions. SVM works on different kernel functions like linear, polynomial and radial functions. These functions are utilized according to the need of situation. The linear kernel function with SVM is suggested and proposed by the researchers. Because in linear kernel the SVM sparse the features and classes or samples and docs in space in linear fashion that help to make the boundary easily in the classes. Thorsten [22] had explained the support vector machine deeply highlighting the evidences proved by SVM practically and theoretically for using it as text classifier. Their work was to text categorization with SVM on multiple features. Since the SVM is proven

algorithm/classifier for specifically text classification and it's the best classifier amongst the latest classifier that's why we will use it as primary classifier in our work. The system diagram for the classification of patient descriptions is shown in figure 1.5.

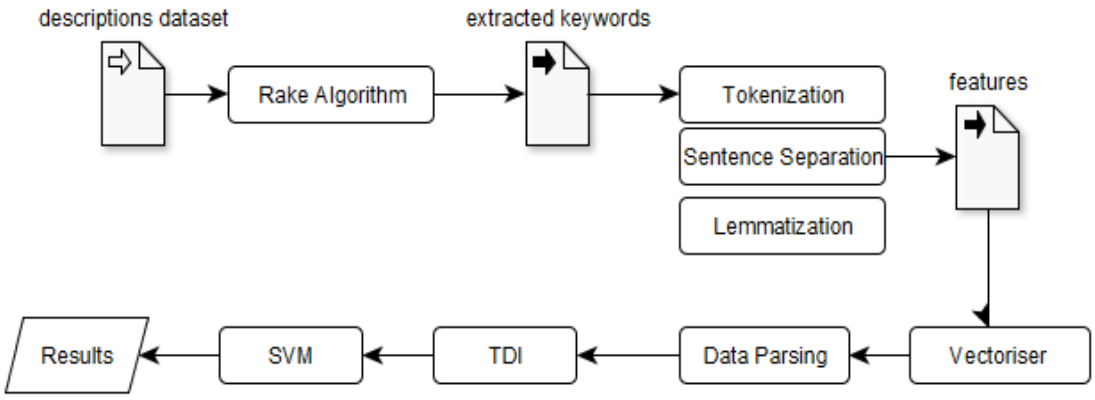

Figure 1.5: System diagram for the classification of patient descriptions

3.3.2 Data Processing

In the data processing there are defined steps that will help to work on the text data sets. The researchers especially in the field of natural language processing has said that the dataset filtering and processing is the main part of the work most of the time. Having said that the patient descriptions were undergone the cleansing and comprehension of the data. The different approaches can be utilized. The classification from the text based patient descriptions has been done. The classification on text can be done with various approaches. These approaches involve the features that can be utilized by the classifiers to do the targeted classification. The conversion of text into features was taken as subject in this phase where the text features are made for the smooth process of classification. As in section 3.2.1.1 the classifier for the patient descriptions has been decided and chosen so the data processing means the features or the key phrases extraction.

3.3.4 Data Set

The data set for this work was made by ourselves. We are working in the context where the patient descriptions are not maintained. There is no mechanism for the digitalization of the patient records in the form of text. There are hospitals and medical centers where they have generalized information of patients and mostly the centers don't share such information. We have tried to find the datasets from the different dataset repositories placed online but can't find the dataset according to our need that is textual patient description that is mostly done by the patient for helping doctors to diagnose further. The problem was that there is misconception of the meaning of patient's descriptions to understand and we are going to solve this problem with the help of NLP and machine learning. To handle and solve this issue of dataset we have formulated, prepared our own dataset that is the patient descriptions in the form of text. This dataset is prepared with the help of few online medical forums like patients.info and I2B2 dataset. The dataset comprises of total 450 records including 300 training records and the 150 testing records. The 300 records and prepared or gathered from the patients.info forum. These records has general discussion of online patients about their disease related problems. They describe their symptoms and other health related issues.

The datasets are comprised of 6 diseases as classes in the case of classification. The disease set is {'Abdominal Issues' , 'Heart' , 'Fever' , 'Diabetes' , 'Asthma' , 'Hypertension'}.

The dataset of 300 records was used as train the model. The other 150 records are taken from the I2B2 [23] dataset repository. These 150 records were used as the testing of the model. These records were used for testing because these records or patient descriptions were naïve to the model and also these records were having the more technical usage of language. This dataset was used for the disease classification and then to devise or suggest the medicine. The training dataset was

then labelled with diseases as class to each of the description/record. These labelled descriptions then used as the multi-class classification with SVM.

3.3.5 Feature Extraction

The features for the dataset are mainly the keywords in the form of bi-gram. These features are kept with their co-occurrence to other features. The features or keywords of descriptions are the main key-phrases used in the plain text of patient descriptions or records. These features are extracted through the RAKE algorithm described in section 2.3. The process how these keywords or features are ranked and scored is the key phrase extraction logic of this algorithm. The count of features of keywords is utilized differently on different cases. The most usage was the top ranked keywords. The results were obtained by using 3-keywords, 5-keywords and 7-keywords. The keywords of 300 records training dataset were kept as features in the .csv file for further processing.

3.3.6 Natural Language Processing

NLP techniques are utilized. Consequently, a review of NLP strategies is given in this area. Inside looking and classifying records, no comprehension of the lexical structure of the content is required. Handling lexical structure, among other language particular qualities, is a piece of NLP. NLP is a wide research area, concentrating on understanding. This exploration region is an expansion to the IR area, since understanding the normal language is a decent beginning stage for more unpredictable IR activities.

NLP have a very strong sub-area morphological and lexical analysis

3.3.6.1 Morphological analysis of Features

The extracted features were undergone or passed through the natural language processing pipeline. This pipeline is based on the different processes related to syntax, semantic, etc. The morphological analysis is shown in figure 1.6.

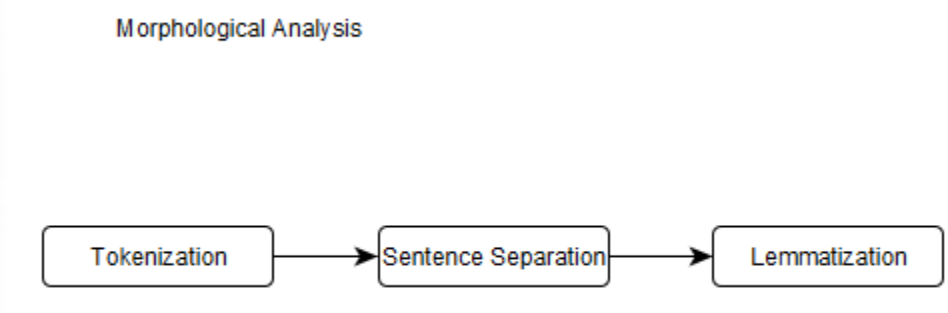

Figure 1.6: Morphological analysis of features

Tokenization: basically splitting a sentence into symbols, words or phrases. In its most simple form, each space and punctuation mark is the starting point of a new token. So the patient descriptions first tokenized into keywords or simple tokens. This technique is the basic and first most to process the data. The patient descriptions were first tokenized into tokens. These tokens also converted into lower case for the better processing of text. The training set of patient descriptions passed through this process.

Sentence Separation: This is spotting the boundaries of a sentence. This is typically done by looking for periods, question marks, exclamation points and capital letters. However, a period is also used in abbreviations, so sentence segmentation is not as straight forward as it sounds. The patient descriptions are comprised of multiple sentences. This helped to assign the boundary. The tokens generated by the tokenizer are then passed through the sentence segmentation. In this phase

the each sentence boundary is identified so that the sentence can play specific role in the further processing. The occurrence of period will separate the sentence from the other part of the text.

Lemmatization: Lemmatization is a more classy method of stemming using dictionaries, which returns the base of the word instead of the stem. For example 'playing' would become 'play'. Notice that 'calling' becomes 'call', which is different from the 'playing' example. Lemmatization is more complex than stemming, because a lot of language knowledge is required to perform this method. The process of lemmatization is quite comprehensive process as compared to stemming also it takes more processing than stemming. In this process the words or tokens are converted to their root words or tokens. The different forms of words with helping verbs and other forms of verbs are converted or lemmatized to the basic root word. We have used the lemmatizer of WordNetLemmatizer.

3.3.7 Feature Vectorization

The classifier and algorithm for the patient descriptions will take the input in the form of vectors for classification, so the textual format of patient descriptions needed to be converted into vectors. The text of patient descriptions is undergone the all steps of natural language processing techniques or strategies for data processing. It includes the NLP processes as pipeline described in section 3.3.6 Natural language processing like tokenization, sentence segmentation and lemmatization. These all are used for the purpose of feature cleansing and to make them in a format that will help to classify.

Feature universe: The features after the lemmatization are sparsed into a universe of features. This universe assign a number to each feature on the basis of occurrence of times, for example it is the total unique features in the patient descriptions of the dataset either training or testing. This

universe helped to make the TDI for the features in the dataset. The length of universe decides the feature vector length or dimension. As the length will be large the vector dimension will be large and result. The universe and the term incidence matrix algorithm is described below:

Algorithm: Universe and TDI

```
Universe = { }
count_of_feature = 0
for data in pre_processed_dataset :
           features= lowercase for x in data
           for f in features :
                 If f not in universe:
                       Universe[x] = count_of_feature
                       count_of_feature ++
tdi = [ ]
vector_length = len(Universe)
count = 0
for data in pre_processed_dataset :
      train_docs = [ 0 ] * vector_length
      features= lowercase for x in data
      for f in features :
            train_docs[Universe[f]] =1
      tdi.append( [ train_docs ,  class_label[count][0] ] )
```

The algorithm for making own term document incidence matrix is defined in above algorithm. The feature universe is made that shows each feature based on its frequency and count. The universe is the main part that helps to make the own term incidence matrix. To make the matrix the length of universe is used to initiate the vector of that length. Then the features are first converted into the lowercase, this is necessary because there can be same features having many forms like the capitalized and non-capitalized start of the feature. This way only the count of feature will be stored to put it into a dimension or to keep it with a number. Then in TDI the all features are sparsed and a value of ‘1’ is assigned to the location where the feature is existing.

3.3.8 Self defined term incidence matrix

We have defined our own term-document matrix with the title 'keyword-description matrix'. This matrix help to further work on the vectoriser that will help ultimately to classification of patient descriptions. Our defined term-document-incidence matrix is represented in the table 1.1

For example we have 3 patient descriptions keywords, these keywords are the last 3 keywords of first three descriptions of 300 dataset as:

- Desc_1={ `rapid heartbeat','feel','spasms' }`
- Desc_2={ `'breathe', 'stomach pain', 'asthma' }`
- Desc_3={ `'feel', 'asthma', 'time',` }

Table 2.1: term-document-incidency

Descriptions	Rapid heartbeat	Feel	Spasms	Breathe	Stomach .	Asthma	Time
Desc_1	1	1	1	0	0	0	0
Desc_2	0	0	0	1	1	1	0
Desc_3	0	0	0	0	0	1	1

The descriptions can have representation of a vector space model, which is a vector representation of a specific description set. Each description has a corresponding vector, and each keyword in the vector corresponds to a term in the description. So each dimension of the created vector corresponds to an index term. The value of the keyword or term will be 1 if it is occurring in the

vector otherwise it will be 0. The description in the form of vector can be represented as an example their dimension will be:

Desc_1 = {1,1,1,0,0,0,0}

Desc_2 = {0,0,0,1,1,1,0}

Desc_3 = {0,0,0,0,0,1,1}

In our case of terms or keywords they are atomic in sense of word or token length, they are in the form of n-grams like the 2-grams when two keywords are used and tri-grams when three keywords are used. So their vector will be the combine force in a direction and the same will be located at the time of query, the vector will be mapped or matched for the disease class identification or for the medicine proposition.

3.4 Query formulation

The query formulation is the main task in medicine suggestion. The query is formulated from the features or keywords of the patient descriptions. The top scored keywords used for this purpose to take the good results. The 3-feature, 5-feature and 7-feature based query is formulated. This query then applied and verified through the trained model of classification. The query predicted the correct disease label and then announced the medicine to the specific disease, here we proposed the one medicine for the prototyping purpose. The list of medicines can be predicted by this system. The query working is simple, the query features are converted into vectors with the same dimensions vector as in the training set of the model. Then this query is given to the model it predicts from its training. The predicted query is then applied to the dataset containing the patient descriptions and medicine labels. The same query vector matched or the takes the services of

trained model to predict the medicine. This can be done with simple strategy of matching of cosine similarity. The query formulation is shown in figure 1.7.

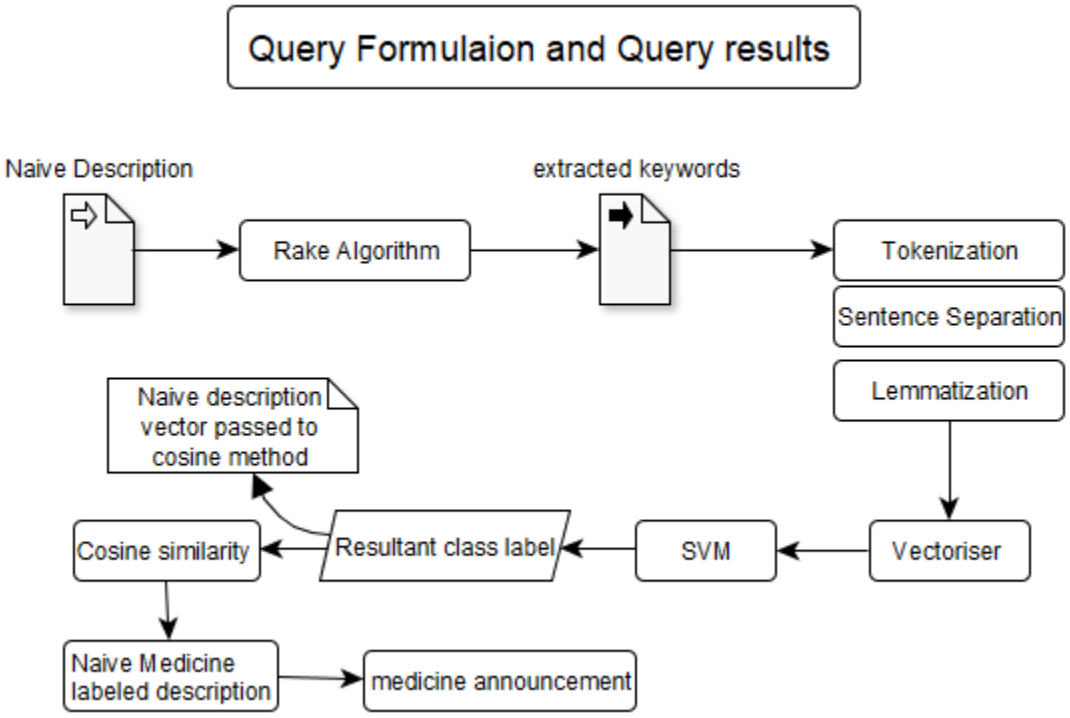

Figure 1.7: Query formulation for the naïve description as test document and results

3.4.1 Cosine similarity

The scipy library have this function to check the two vectors either they are maximum matched or not. The query vector matched with each other vector in the dataset of patient description having medicine labels. The vector which have maximum angle is less matched and the vector in the same direction with smallest angle is the maximum matched vector.

$$\cos\theta = \frac{a.b}{|ab|} \qquad (8)$$

In equation (8) a and b are the vectors, their cosine angle decide the similarity level. If the value of ‘ θ ‘ is 0 then the similarity is maximum and if the angle is 90 then similarity will be minimum equal to 0. On the basis of this similarity level the medicine is proposed, each vector has medicine label at the last index of the vector. The most similar vector will return the medicine label after calling this process of medicine proposition.

3.4.2 Algorithm for medicine proposition

There are mainly two datasets Da and Db. The Dx is naïve dataset for testing

Da=[descriptions, disease_label]

Db=[descriptions, disease_label, medicines]

Tr → Using Da as Training models

Dx =[descriptions]

Using Tr for classifying the Dx and then applying label of disease.

Cosine (distance (Dx, Db)) → Medicine to suggest

Also applied Tr on Db to validate our model.

The vector or file of (Da: 450.csv) was used to train the SVM then the new description was tested with same universe of 450.csv. Now the new description named xdoc was vectorized with the universe of 491 that is equivalent to the (Db: medicinelabelvectorized150.csv) file. In this way after getting the label of disease we have searched that labelled_bucket in medicine file. The result of medicine announced of that vector which is maximum matched to the xdoc

Chapter 4

4- RESULTS AND ANALYSIS

In this chapter the result of the system are discussed and their evaluation and analysis is described. The precision and recall was good of multi-class classifier model that is precision 0.99, recall 0.98 and f1-score is 0.98. The results on the naïve dataset of descriptions with medicine label is precision 0.86, recall 0.81 and f1-score is 0.81.

In text classification the classifier algorithm uses the training dataset for its training in supervised learning. The classifier build the model from training patient descriptions. The model then used to label the unclassified documents in our case the patient descriptions. For the evaluation purpose the classifier is passed on to the test dataset for testing purposes of developed model. The test dataset contains the labelled documents and the original labels are then removed from the test dataset. The classifier mark each document with label learned during training, then these marked documents are compared with the original labels of the test dataset for the accuracy and other performance measures.

There are methods that can be utilized as the performance measures for classifiers in terms of effectiveness and efficiency. The effectiveness can be measured by employing the recall method. **Recall** In classification problems the recall is measured with few standard terms like true positive false positive, true negative and false positive. Positive and negative are the terms mostly used by the researchers with meaning that these are the predictions of the classifier.

The formula to represent recall is

$$\text{Recall} = \frac{tp}{tp+fn} \qquad (9)$$

In equation (9) the '*tp'* is true positive values and '*fn'* is the false positive. So the ratio of true positive with addition of true positives and false negatives is the recall of the document.

Precision is also measure with same parameters as recall is measured. The precision for a class of classification problem is the number of *'tp'* true positive values its means the items or documents correctly marked true which were originally from positive class divided by the total number of documents which are labelled with any label in the positive class, its means the sum of true positive and false positive.

The formula to represent precision is

$$\text{Precision} = \frac{tp}{tp+fp} \qquad (10)$$

In classification tasks, the precision and recall score matters for example the 1.0 score means that the classifier marked the label correct and the label matched the original label. In other words an item or document marked by classifier with class X and the item originally was also belonging to the class X.

Fi-score it comes under the F-measure, the f1-score is also known as the balanced f-score.it is the harmonic mean of the precision and recall, this is closed to their average of precision and recall that's why called harmonic mean. This is also called f1-measure because in this case the both precision and recall equal weight.

The formula to represent f1-score is

$$\text{F1-score} = 2.\frac{precision\,.\,recall}{precision+recall} \qquad (11)$$

4.1 Classification results

The classification is done on both datasets. The dataset to test the model and the naïve dataset for the medicine proposition. Since the results got by using the classifiers with sequence of features like 2-feature, 3-feature, 5-feature and 8-featiure results. The results on smaller dataset were not good but after the dataset enhancement results has been improved.

4.1.1 Accuracy of Bi-Class & Tri-Class with three keywords per description

(SVM=support vector machine, DT=Decision tree, RF=random forest, AB=AdaBoost)

	Diabetes	**Hypertension**	**Heart Problems**
Asthma	SVM: 0.56; DT: 0.56; RF: 0.54; AB: 0.53	SVM: 0.61; DT: 0.55; RF: 0.53; AB: 0.57	SVM: 0.60; DT: 0.62; RF: 0.64; AB: 0.64
Diabetes		SVM: 0.62; DT: 0.62; RF: 0.51; AB: 0.60	SVM: 0.67; DT: 0.63; RF: 0.53; AB: 0.60
Hypertension			SVM: 0.66; DT: 0.63; RF: 0.57; AB: 0.66

Table 1.2: Results of Bi-Class & tri-Class sequence

In Table 1.2 the results are calculated by the combination of two classes and three classes. Like the results of 'Asthma' combined with 'Diabetes', 'Hypertension' and then with 'Heart problems'.

	Fever	**Abdominal issues**
Heart Problems	SVM: 0.67; DT: 0.65; RF: 0.63; AB: 0.70	SVM: 0.62; DT: 0.62; RF: 0.68; AB: 0.62
Fever		SVM: 0.68; DT: 0.66; RF: 0.66; AB: 0.65

Table 1.3: Results of Bi-Class & tri-Class sequence

4.1.2 Results of multi-Class Testing: 5-keywords per description

Table 1.4: Multi-class results on 5-keywords

SVM	**Random Forest**	**Decision Tree**	**AdaBoost**
[0.93939394 0.9375 0.96875 .96774194 0.96774194 .96774194 0.93333333 .93103448 0.96428571 0.92592593] **Average Accuracy** **0.950344920215**	[[0 0 0 ..., 0 0 0] [0 0 0 ..., 0 0 0] [0 0 0 ..., 0 0 0] [0 0 0 ..., 36 2 1] [0 0 0 ..., 0 77 1] [0 0 0 ..., 0 0 40]] **Average Accuracy** **0.926559957392**	[[0 0 0 ..., 0 0 0] [0 0 0 ..., 0 0 0] [0 0 0 ..., 0 0 0] [0 0 0 ..., 35 0 2] [0 0 0 ..., 1 75 0] [0 0 0 ..., 0 0 36]] **Average Accuracy** **0.93573644103**	[[0 0 0 ..., 0 0 0] [0 0 0 ..., 0 0 0] [0 0 0 ..., 0 0 0] [0 0 0 ..., 19 0 11] [0 0 0 ..., 0 76 0] [0 0 0 ..., 0 0 31]] **Average Accuracy** **0.811427243193**

In figure 1.8 the results of different classifiers like SVM, Decision tree, Random forest and Adaboost are shown. These results are calculated with the different features like 3-features, 5-features and then 8-features. As the features are increased the results of classifiers outperformed the previous results.

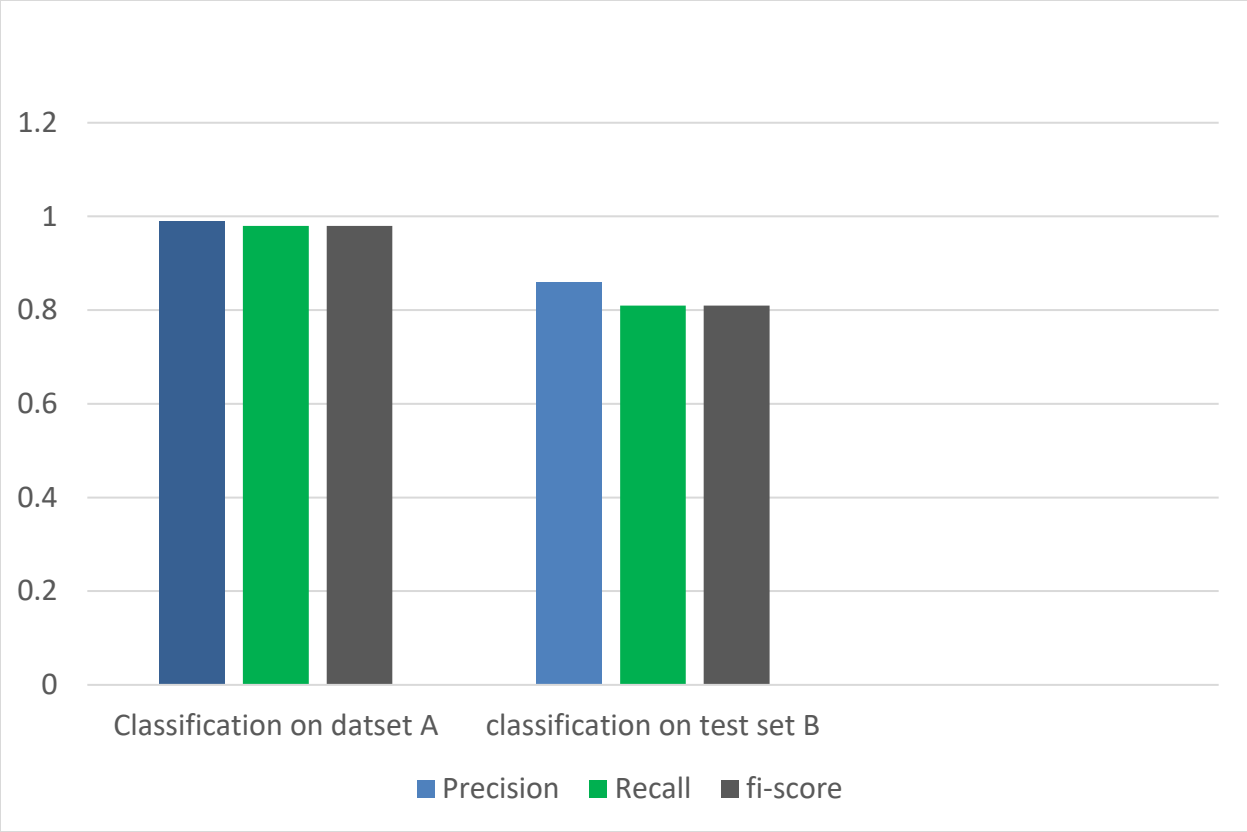

Figure 1.9 : precision , recall and f1-score of testing of both datasets

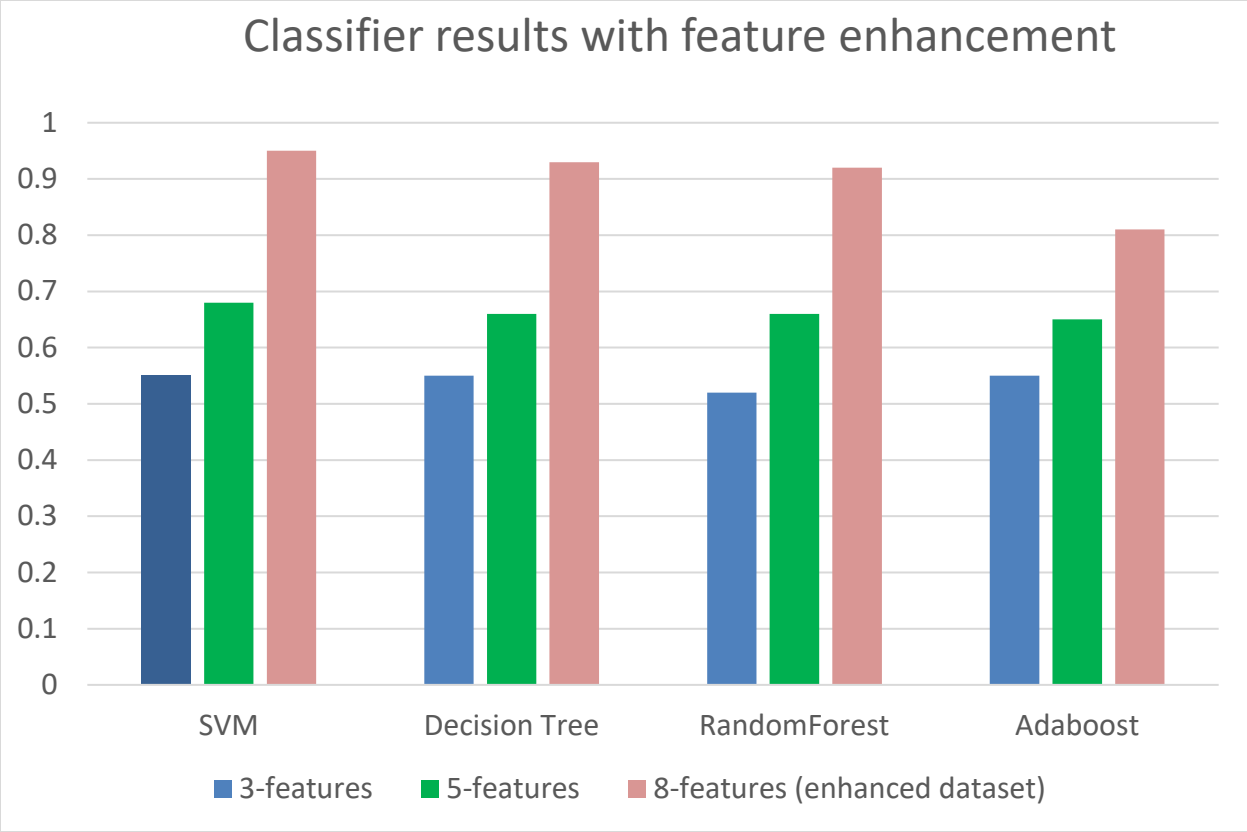

Figure 1.8: Classifier results with feature increment

In figure 1.10 the results of enhanced dataset has shown. All the measuring parameters performed well. The recall shows that except the 'Asthma' disease all other disease have been assigned correct label.

	precision	recall	f1-score	support
Abdominal issues	1.00	1.00	1.00	11
Asthma	1.00	0.90	0.95	10
Diabetese	1.00	1.00	1.00	9
Heart Problems	1.00	1.00	1.00	16
Hypertension	1.00	1.00	1.00	8
fever	0.88	1.00	0.93	7
avg / total	0.99	0.98	0.98	61

Figure 1.10: Results of multi-Class: On enhanced dataset & 8-keywords per description

4.1.3 Query Results: Multi-Class classification

	precision	recall	f1-score	support
Abdominal issues	1.00	1.00	1.00	14
Asthma	0.55	1.00	0.71	6
Diabetese	0.92	1.00	0.96	11
Heart	0.93	0.65	0.77	43
Hypertension	0.58	0.88	0.70	8
fever	0.60	0.86	0.71	7
avg / total	0.86	0.81	0.81	89

Figure 1.11: Query on multi-class testing : On enhanced dataset & 8-keywords per description

Chapter 5

5- Conclusion

In medical field mostly the problems need their solutions for the betterment of the society in broader level. The natural language processing can help many things related to text. The patient descriptions in our local context are written in the form of textual format. In this thesis study we have developed a solution for the medical practitioners and the doctors (in future these can be replaced by the robot doctors or virtual doctors). Our solution is more focused towards the processing of text and feature extraction from the plain text and then to form a query that can work both for the classification of the textual descriptions and to suggest the medicine based on the information given in description. We have employed the patient descriptions for this purpose and applied the natural language processing and machine learning techniques to provide the first aid

type decision to proceed for further diagnosis. We have developed a system that takes the patient descriptions in textual format and then process it as described in the methodology section. We have got good results with small own developed dataset. Also we have calculated the results of multiple features like 3_features, 5_features and 8_features. The change of sequence of classes has also produced good results. Like heart and hypertension when tested 2_classes at a time their results were good. We have also tested that the dataset enhancement from 300 to 450 patient descriptions, this has increased the performance of the model. We got precision and recall 0.99 and 0.98 respectively on 300 patient descriptions dataset. After the training of the model we tested our model on 150 naïve patient descriptions, though these descriptions were more technically rich, the results of precision and recall on this dataset were 0.87 and 0.81 respectively.

Same time we have also worked on the query retrieval side that is done on our handmade test bed. The retrieval is done in a way that the dataset of naïve patient descriptions was labelled with medicines names against each description. After the description marked with correct label of disease, the patient descriptions vectors of all this marked disease bucket are compared/matched using the cosine similarity, then most matched vector/description medicine label announced as a suggestion of medicine. Now the proposition of medicine accuracy and validity depends on the perfect marking of the description with disease label. As the recall of the classifier is good the suggested medicine will be accurate one. In our work it was problematic to find the text based patient descriptions dataset of different diseases, that's why we have developed our own dataset of patient descriptions from online patient health discussion forums. In future this work can be done with more good results by utilizing the Named Entity Recognition and ML with word embedding. Also this work can be extended in the chat bot form with the large dataset by extracting ML features from the text.

6- References

[1] Djulbegovic, Benjamin, and Gordon H. Guyatt. "Progress in evidence-based medicine: a quarter century on." *The Lancet*(2017).

[2] Sarker, Abeed, Diego Molla, and Cecile Paris. "Automated text summarisation and evidence-based medicine: A survey of two domains." *arXiv preprint arXiv:1706.08162* (2017).

[3] Guo, Xiaoxiao, et al. "Learning to Query, Reason, and Answer Questions On Ambiguous Texts." (2016).

[4] Mollá, D., Santiago-Martínez, M.E., Sarker, A. et al. Lang Resources & Evaluation (2016) 50: 705. https://doi.org/10.1007/s10579-015-9327-2

[5] http://www.jfponline.com/ [accessed on 16-10-2017]

[6] Dönmez, İlknur, and Eşref Adali. "Extracting phrase-content pairs for Turkish sentences." *Application of Information and Communication Technologies (AICT), 2015 9th International Conference on*. IEEE, 2015.

[7] Chandurkar, Avani, and Ajay Bansal. "Information Retrieval from a Structured Knowledge Base." *Semantic Computing (ICSC), 2017 IEEE 11th International Conference on*. IEEE, 2017.

[8] Kim, Su Nam, et al. "Automatic classification of sentences to support evidence based medicine." *BMC bioinformatics* 12.2 (2011): S5.

[9] Sarker, Abeed, Diego Mollá, and Cecile Paris. "Query-oriented evidence extraction to support evidence-based medicine practice." *Journal of biomedical informatics* 59 (2016): 169-184.r.

[10] Yu, Hong, and Yong-gang Cao. "Automatically extracting information needs from ad hoc clinical questions." *AMIA annual symposium proceedings*. Vol. 2008. American Medical Informatics Association, 2008.

[11] Pratt, Wanda, and Lawrence Fagan. "The usefulness of dynamically categorizing search results." *Journal of the American Medical Informatics Association* 7.6 (2000): 605-617.

[12] Cao, Yonggang, et al. "AskHERMES: An online question answering system for complex clinical questions." *Journal of biomedical informatics* 44.2 (2011): 277-288.

[13]- Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, Dan Jurafsky and James H.Martin, Draft of August 28, 2017.

[14] - Zhai, Chengxiang, and John Lafferty. "A study of smoothing methods for language models applied to ad hoc information retrieval." ACM SIGIR Forum. Vol. 51. No. 2. ACM, 2017.

[15] Hulth A 2004 Combining machine learning and natural language processing for automatic keyword extraction. Stockholm University, Faculty of Social Sciences, Department of Computer and Systems Sciences (together with KTH).

[16] - Rose, S., Engel, D., Cramer, N. and Cowley, W. (2010) Automatic Keyword Extraction from Individual Documents, in Text Mining: Applications and Theory (eds M. W. Berry and J. Kogan), John Wiley & Sons, Ltd, Chichester, UK. doi: 10.1002/9780470689646.ch1

[17] Conway, Mike, et al. "Classifying disease outbreak reports using n-grams and semantic features." International journal of medical informatics 78.12 (2009): e47-e58.

[18] Fatima, M. and Pasha, M. (2017) Survey of Machine Learning Algorithms for Disease Diagnostic. Journal of Intelligent Learning Systems and Applications , 9, 1-16

[19] Scott, S., & Matwin, S. (1999, June). Feature engineering for text classification. In *ICML* (Vol. 99, pp. 379-388).

[20] Kumari, V. Anuja, and R. Chitra. "Classification of diabetes disease using support vector machine." *International Journal of Engineering Research and Applications* 3.2 (2013): 1797-1801.

[21] Chaurasia, Vikas, and Saurabh Pal. "Data mining approach to detect heart diseases." (2014).

[22] Joachims, Thorsten. "Text categorization with support vector machines: Learning with many relevant features." *European conference on machine learning*. Springer, Berlin, Heidelberg, 1998.

[23] "Deidentified clinical records used in this research were provided by the i2b2 National Center for Biomedical Computing funded by U54LM008748 and were originally prepared for the Shared Tasks for Challenges in NLP for Clinical Data organized by Dr. Ozlem Uzuner, i2b2 and SUNY."

[24] Lewis, David D. 1992a. An evaluation of phrasal and clustered representations on a Text Categorization Task.*SIGIR-92*. 37-50

[25] Névéol, Aurélie, et al. "Clinical Natural Language Processing in languages other than English: opportunities and challenges." *Journal of biomedical semantics* 9.1 (2018): 12.

7-Glossary

Appendix-A

EBM= Evidence based medicine is a technique in which the decision is taken for the health care of individuals by employing the best available evidences.

QRAQ= Query, Reason, and Answer Questions is an agent based system that learns and answer the user questions.

RL=Reinforcement Learning is technique in which the agent learns by asking the missing information.

NLP=Natural Language Processing is an Artificial Intelligence sub-area focuses on removing the ambiguity from the texts and speech.

Word-Net = word-net is a Lexical database of English Language

POS= Part of Speech is tagger that is used to specify parts of speech in a sentence.

CRF=Conditional Random Field is a classifier type statistical model that is used for pattern recognition and in machine learning.

UMLS=Unified Medical Language Systems is corpus of files and softwares that combinely works for standards of computer systems.

QSpec= Query Specific is a system that focusses on the query part.

DynaCat=Dynamical Categorisation is a system that categorises the results of queries to corpus dynamically.

JFP= The Journal of Family Practice is a place where all types of medical Q/As data sets are there.