Developing Disease Classification System based on Keyword Extraction and Supervised Learning

Muhammad Suffian, Muhammad Yaseen Khan, and Shuakat Wasi
Department of Computer Science,
Mohammad Ali Jinnah University,
Karachi, Pakistan

***Abstract*—The Evidence-Based Medicine (EBM) is emerged as the helpful practice for medical practitioners to make decisions with available shreds of evidence along with their professional expertise. In EBM, the medical practitioners suggest the medication on the basis of underlying information of patients descriptions and medical records (mostly available in textual form). This paper presents a novel and efficient method for predicting the correct disease. Since these type of tasks are generally accounted as the multi-class classifying problem, therefore, a large number of records are needed, so a large number of records will be entertained in higher n-dimensional space. Our system, as proposed in this paper, will utilise the key-phrases extraction techniques to scoop out the meaningful information to reduce the size of textual dimension, and, the suite of machine learning algorithms for classifying the diseases efficiently. We have tested the proposed approach on 6 different diseases i.e. Asthma, Hypertension, Diabetes, Fever, Abdominal issues, and Heart problems over the dataset of 690 patients. With key-phrases tested in the range [3,7] features, SVM has shown the highest (93.34%, 95%) F1-score and accuracy.**

***Keywords*—Natural language processing; Machine Learning; Multi-Class Classification; Patient descriptions; Keyword Extraction**

I. Introduction

The idea of Evidence-Based Medicine (EBM) caused incredible enthusiasm among well-being experts. As indicated by definition [1] Evidence-Based Medicine is the medication suggested by the doctors underlying the available health status of the 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. 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 revolutionised the EBM concept. This mechanism is helpful for the doctors to pick the latest curing medications for the severe type of diseases. In spite of many hurdles, Evidence-Based Medicine practice has gained the reputation over recent years due to the reasons, like the improvements in patients' health-care. Research advancements are removing the barriers in EBM and it is inferred that the boom will come with NLP techniques. Our problem is an inspiration from Sarker et al. work [2]. They discussed the problems and obstacles in evidence-based medicine faced by the practitioners. They categorised 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. Névéol et al. [3] 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 second thing is to classify the patient description into a specific disease. 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 [1]. 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 and classification of patient description 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.

The rest of the paper is organised as follows: In section II the related work is highlighted, in section III the methodology and experimental setup is discussed. The section IV provides information regarding to the data sets used in the experiment. In section V the results are shown and discussed. in the end conclusion and future work are presented in section VI and VII respectively.

II. Related Work

An approach similar in spirit to our work is discussed and modelled in QRAQ [4]. The authors discuss 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 deduces the variables from the

fact in the problem. Secondly, if the 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 [5] Molla et al. built a corpus for the text processing. They have taken the data set from the clinical inquiries segment of the journal dealing with family practice [6]. They annotate the data using the annotation techniques like automatic extraction, manual annotation and the rephrasing text. The inquiry sentence is used as the query and the retrieval text then summarised to answer. The summary of the 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 are based on the score of matching to the evidence.

In the work of Molla et al. [5] one thing can increase the accuracy of the retrieved summaries that is the removal of ambiguity from the input sentence/query. In [7] 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 [8]. These phrase-concepts pairs like syntactic and semantic information of sentences have shown with matrix representation.

Avani et al. showed an Question/Answering system [9] which 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 [10] this determines the type of the question and also the category of the 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 categorise the questions in which there is a call for action. They evaluated their Question/Answer system on TREC 2004 question data set. In [10], Kim et al. build a sentence classifier that 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 meta-thesaurus from UMLS (Unified Medical Language Systems) was used, then directly query the thesaurus with each input token. MetaMap analyser 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 utilised on results after querying, the ambiguity of query and question meaning before applying on data set is not handled in their work.

Sarker et al. presented a query focused approach for text summarisation to support evidence-based medicine [11]. 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 summarisation performance. They classified the questions in their corpus into medical topics using the approach [12]. For the better question associations with summaries, they set two semantic types for each question (a) important question semantic types that were identified during training and (b) important answer semantic types that are identified from human-authored summaries in training. They evaluated their approach using ROUGE evaluation tool, their QSpec system outperforms previous systems working on the 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 summarisation task.

In [13] Pratt et al. gave a new approach for categorising the search results was implemented with the name DynaCat system. In this, they divided the semantics of dynamic categorisation 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 the 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 generalises the query into the specific category like categorising 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 the 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 it did not assign the category on the basis of disease.

In [14] Cao et al. 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 Information retrieval (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.

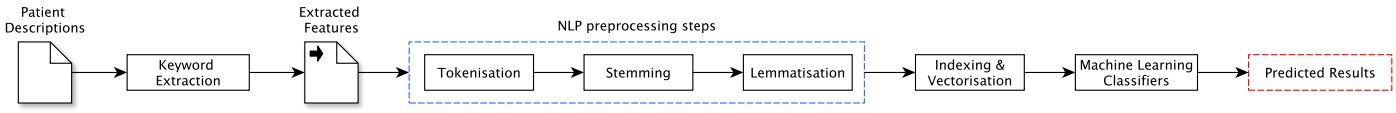


Fig. 1. Architectural scheme of query formulation and disease classification from patient descriptions.

III. Methodology

The proposed methodology has two straight forward phases. The first phase is related to the extraction of specific key-phrases out of detailed patient descriptions/records, and application of standard NLP techniques on the extracted information followed by indexing and vectorisation of features. In the second phase, these key-phrases are then employed for the supervised learning and disease classification. These two phases are described in sections III-A and III-B respectively. The figure 1 illustrates the pipeline of the methodology and the details of both phases are given is two separate sections accordingly.

A. Extraction, Preprocessing & Pre-classification tasks

Keywords and keyphrases extraction. The keywords (or extended keywords i.e. key-phrases) are the central representatives of the content in any document. It helps to identify the basic theme of any document. Hence, in spite of analysing and computing the whole bunch of documents for formulating queries, it is more easy-to-use to extract the important keywords and proceed with the rest of computational procedures in comparatively space and time efficient way. In this regard, Rose et al. [15] presented the idea for keyword extraction in which they described and compared their algorithm, Rapid Automatic Keyword Extraction (RAKE), with different NLP based methodologies and algorithms with their specific use. RAKE is an unsupervised, domain-independent algorithm that works on co-occurrence graph, it extracts the candidates for the key phrases and words from the text, then checks whether these can be declared keywords or not and then the score is assigned to each keyword. The scoring metric is quite simple [15]. Letting w be the word in a corpus, the score will be a ratio i.e. $deg(w)/freq(w)$, where $deg(w)$ is the degree of word w and $freq(w)$ is the frequency of the word w in the given corpus. We can say the keywords extracted through RAKE are the features that are mostly in the form of n-grams where $n > 1$. In our experiments, we took the top $3-7$ extracted features for training and classification. Table I provides the example of extracted keywords and their scores accordingly.

TABLE I. Sample of Patient Description and Extracted Features by RAKE

Sample Patient Description
This is a 36-year-old woman with a history of type-I, chronic renal insufficiency on hemodialysis as well as chronic skin ulcers who was at hemodialysis on the day of admission when she developed high temperature to 101, chills, and rigors.
Extracted Features/Key-phrases
('chronic renal insufficiency', 9.0), ('chronic skin ulcers', 9.0), ('developed high temperature', 9.0), ('36-year-', 1.0), ('woman', 1.0), ('history', 1.0), ('type-', 1.0), ('hemodialysis', 1.0), ('day', 1.0), ('admission', 1.0), ('chills', 1.0), ('rigors', 1.0), ('101', 0)

Preprocessing. The core and essential part of any task in the domain of data-sciences is preprocessing. With the extracted features in the previous phase, regular methods of case-folding, lemmatising and stemming are applied. These methods are supposed to sort words so as to group together inflected or variant forms of the same words. These methods are employed by using the Natural Language Toolkit (NLTK) module in Python [16]. The extracted features are tokenised prior to pass through these methods.

Indexing and vectorisation of feature vectors. The classifying ML algorithms require input in a vector format. Thus, at this stage, the main goal is the transformation and vectorisation of extracted features. These vectors are typically a boolean representation of the documents in an n-dimensional space, where each term resides at a separate dimension. Thus, if a term t_i is present in the document d_j, the vector v_j representing the document d_j will mark 1 at the index corresponding to the term t_i, otherwise, there will be 0 representing the absence of the term in the document. The collection of these vectors is a matrix and often named as term-document incidence (TDI). Table II renders an example of documents in a vector space model.

TABLE II. Example of Term-Document Incidence

	t_1	t_2	t_3	$\cdots$	t_n
d_1	1	0	0	$\cdots$	1
d_2	1	1	0	$\cdots$	0
d_3	0	1	1	$\cdots$	1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
d_n	1	0	1	$\cdots$	1

In order to construct TDI for extracted features, we created a universe of features ($\mathbb{U}$) where each tokenised term t is tagged with a unique index number. For this kind of tasks dictionary (as the data-structure) is the most suitable solution. Algorithm 1 will give the simple and robust solution for universe construction, where the tokenised features are check iteratively in the dictionary for their existence, if the result of

Result: A dictionary ($\mathbb{U}$) with the terms as keys and respective index number as values.
$\mathbb{D} \leftarrow$ be the set of extracted features;
$\mathbb{U} \leftarrow$ be the empty dictionary;
$\mathbb{C} \leftarrow 0$;
for *each document* d *in* $\mathbb{D}$ **do**
  $\mathbb{T} \leftarrow$ split d into tokens;
  for *each token* t *in* $\mathbb{T}$ **do**
    if $t \notin \mathbb{U}$ **then**
      $\mathbb{U}_{[t]} \leftarrow \mathbb{C}$;
      $\mathbb{C} \leftarrow \mathbb{C} + 1$;
    end
  end
end

Algorithm 1: Algorithm for building universal set of distinct terms.

lookup is false (i.e. $t \notin \mathbb{U}$) then the term t is added to the $\mathbb{U}$; where the term t is set to the key with value $\mathbb{C}$ as a key-value pair, along with this process the counter $\mathbb{C}$ gets increment by 1 for serving as the index of forthcoming term.

Once $\mathbb{U}$ is created we can utilise the dictionary to proceed towards the construction of term-document incidence. Algorithm 2 shows the simple procedure, where the collection of preprocessed documents i.e. patient descriptions as $\mathbb{D} = \{\langle d_1, l_1, m_1\rangle, \langle d_2, l_2, m_2\rangle, \cdots, \langle d_n, l_n, m_n\rangle\}$ (where d_i is the extracted features, l_i is the label/class, and m_i is the medication accordingly) is going to be vectorised with respect to $\mathbb{U}$, and $\mathbb{I}=\{v_1, v_2, v_3, \cdots, v_n\}$ is an empty list in which all vectors have to be appended such that v_i is the corresponding vector representation of d_i. Letting X be the local list of zeros equal to size of $\mathbb{U}$. Iteratively each document is split into the set of tokens and each token t is looked up in $\mathbb{U}$ that gives the index-value stored agains t, hence, the 1 will be replaced at the index $\mathbb{U}_t$ in the X. Technically in the end of procedure the correctness can be checked through $|\mathbb{I}|=|\mathbb{D}|$, and since, $\mathbb{I}$ is a non-sparse matrix/list of lists, therefore, the length of vector is equal to the size of dictionary $|v_i|=|\mathbb{U}|$.

Result: A non-sparse matrix showing the boolean representation of documents in n-dimensional vector space.
$\mathbb{U} \leftarrow$ be the dictionary having terms as keys and index numbers as values (generated through algorithm 1);
$\mathbb{D} \leftarrow$ be the set of extracted features;
$\mathbb{S} \leftarrow$ be the size/length of $\mathbb{D}$;
$\mathbb{I} \leftarrow$ be the empty list;
$\mathbb{C} \leftarrow 0$;
for *each document* d *in* $\mathbb{D}$ **do**
  X $\leftarrow$ be the local list of $\mathbb{S}$ zeros;
  $\mathbb{T} \leftarrow$ split d into tokens;
  for *each token* t *in* $\mathbb{T}$ **do**
    X$[\mathbb{U}_{[t]}] \leftarrow 1$;
  end
  $\mathbb{I}.append([\text{X}, label_d])$
end

Algorithm 2: Algorithm for constructing Term-Document Incidence using dictionary defined in algorithm 1.

B. Supervised Learning and Disease Classification

In a machine learning classification system, documents or text with already tagged class labels are set as an input to the ML algorithm that learns the underlying information and patterns from the data to build a predictive model. The document (D) typically consist of features (f_i) and the expected target outcome is a member of the discrete classes (Y) $\therefore D = \{f_1, f_2, f_3, \cdots, f_n\} \rightarrow y_j$, where $y_j \in Y$. Thus, if there are two possible answers (i.e. $|Y| = 2$) then we can say the problem is binary or binomial classification whereas, if the possible answer is more than two (i.e. $|Y| > 2$) then, it would be a multi-class or multinomial classification problem. Hence, the problem addressed in this paper is the multi-class or multinomial classification as there are 6 possible prediction outcomes (Asthma, Hypertension, Diabetes, Fever, Abdominal issues, and Heart problems). The phase in which learning for the predictive model is made is also called training phase. This predictive model is used to classify the unseen data.

In our experiment, we have used 4 different ML algorithms: (a) Random forest (RF) [17] which are counted as the ensemble learning approach in classification. (b) Iterative Dichotomiser 3 (ID3) [18] which is the classifying algorithm that works as a decision tree, (c) Support vector machine (SVM) [19] which is the linear model of classification where data is split into distinct parts in such a way that it holds maximum margin among the splits, and (d) Naïve Bayes, which is a likelihood-based probabilistic classifying function. These algorithms are employed by using the scikit-learn module for Python [20].

We have learned in the previous section about the indexing of extracted features. Utilising the indexing method, we rigorously repeated the experiment with top 3-7 extracted features contributed by RAKE.

IV. Data Set

Availability of relevant data for EBM is a real obstruction. There is as such no evident mechanism for the digitalisation of the patient descriptions/ records in the form of text. Although, there are hospitals and medical centres where they have gathered information about patients but, ordinarily these centres do not share information with the groups who are intended to conduct research in the current domain. Thus, to handle and solve this issue of dataset we have 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 forum like patients.info [21][1] and i2b2 dataset[2] [22], [23].

The dataset comprises of total 240 records from patient.info, and 450 records from the i2b2 dataset. Thus, there are in total 690 patient records. The datasets are comprised of 6 diseases as classes in the case of classification. They are Abdominal issues, Heart, Fever, Diabetes, Asthma, Hypertension. Table III will give you the details of class distribution with respect to both of the datasets. For the sake of training-testing split, we randomised the records and set cross-validation for 10 folds. Hence, the algorithm will use 9 parts of 10 splits in training and remaining will be utilised for the test.

TABLE III. Distribution of Diseases/Classes with Respect to the Datasets

Dataset	Asthma	Hyper-tension	Diabetes	Fever	Abdominal Issues	Heart
patient.info (dataset A)	40	40	40	40	40	40
i2b2 (dataset B)	53	61	60	50	76	150

V. Results

A. Model Evaluation

In this experiment, results are evaluated on the metrics of precision (P), recall (R), accuracy (A) and F1-scores (F). In a conventional binary or binomial classification system, these metrics are calculated with the number of 'true positives' (tp) and 'true negatives' (tn) which means the classifier respectively predicts the instances *positive* that are actually positive, and *negative* that are actually negative. With these two statistics, there are two more i.e. 'false positive' (fp) and

[1] https://patient.info/
[2] https://www.i2b2.org/NLP/DataSets/

'false negative' (fn), which means the classifier mistakenly predicts a negative instance as *positive*, and a positive instance as *negative* respectively. Thus the equations for calculating these metrics in a binary classification system are given below:

$$\mathrm{P} = \frac{tp}{tp+fp} \qquad \mathrm{R} = \frac{tp}{tp+fn} \qquad \mathrm{F} = 2 \cdot \frac{\mathrm{P} \cdot \mathrm{R}}{\mathrm{P}+\mathrm{R}}$$
$$\mathrm{A} = \frac{|\text{correctly predicted instances}|}{|\text{all instances}|} \tag{1}$$

Thus, w.r.t the equations for precision and recall we can say precision as positive predictive rate and recall as true positive rate. An ideal classifying system should have both high precision and recall. While, F1-scores is a harmonic mean between precision and recall, and accuracy shows the overall success of the system. Since, this paper deals with the multi-class problem therefore, the equation for metrics are altered as per the following equations:

$$\mathrm{P}_i = \frac{M_{ii}}{\sum_j M_{ji}} \qquad \mathrm{R}_i = \frac{M_{ii}}{\sum_j M_{ij}} \tag{2}$$

Where, M is a $k \times k$ dimensional matrix such that $k = |\text{classes}|$, and i represents a certain class to be classified. Hence, M_{ii} is the number of tp instances for the class i. Similarly, $\sum_j M_{ji}$ is the aggregate of all values for class i in j^{th} column in M. Whereas, $\sum_j M_{ij}$ is vice-versa of $\sum_j M_{ji}$. Thus, the equations in 2 shows the precision and recall for the class i. The precision and recall for the whole multi-class system will be an aggregate of individual precisions and recalls with respect to all classes and it can be calculated as per equation 3.

$$\mathrm{P} = \frac{1}{k} \sum_{\text{i=0, j=0}}^{k} \frac{M_{ii}}{\sum_j M_{ji}} \qquad \mathrm{R} = \frac{1}{k} \sum_{\text{i=0, j=0}}^{k} \frac{M_{ii}}{\sum_j M_{ij}} \tag{3}$$

B. Experimental Results

Results are quite exciting and interesting. Factually, we witness the improvement in all results with the increment of keywords features form 3 to 7. Tables IV shows the results that on averages, patient.info shows $\approx$15% improvement in F1-score when keyword feature size moved to top-5 from top-3, and further $\approx$12% improvement when feature size updated from top-5 to top-7. Similarly, on the i2b2 dataset, there is $\approx$12% improvement when feature size is moved to top-5 features from top-3, and further $\approx$5% improvement on increasing feature size up to top-7 features. In comparison to the F1-scores, table V shows the improvement in average accuracies on keyword increment. Collectively, patient.info outperforms the results by yielding $+5.75\%$ and $+9.55\%$ difference in accuracy and F1-scores respectively.

TABLE IV. RESULTS OF AVERAGE F1-SCORES W.R.T THE KEYWORDS SIZE AND IMPROVEMENT

Dataset	Average F1-scores			Improvement	
	Top-3 keywords	Top-5 keywords	Top-7 keywords	Keywords 3→5	Keywords 5→7
patient.info	60.50	75.79	88.20	15.29	12.41
i2b2	61.25	73.75	79.40	12.50	5.65

TABLE V. RESULTS OF AVERAGE ACCURACIES W.R.T THE KEYWORDS SIZE AND IMPROVEMENT

Dataset	Average Accuracies			Improvement	
	Top-3 keywords	Top-5 keywords	Top-7 keywords	Keywords 3→5	Keywords 5→7
patient.info	61.25%	80.25%	90.00%	19.00	9.75
i2b2	53.50%	71.25%	76.50%	17.75	5.25

TABLE VI. RESULTS OF DIFFERENT MATRICES ON DATASETS WITH TOP-3 FEATURES

	patient.info (dataset A)			i2b2 (dataset B)		
Algorithm	Prec.	Rec.	F1-score	Prec.	Rec.	F1-score
SVM	79	64	62	94	53	62
Random Forest	61	62	60	90	54	61
Decision Tree	74	61	60	80	57	60
Naïve Bayes	90	57	60	94	52	62

Tables VI, VII, and VIII show the details of precisions, recalls, and F1-scores of the algorithms on top $3-7$ keywords extracted on patient.info and i2b2 respectively. Overall in the entire experimental suit, the average lowest F1-score is $\approx$60 which is shown with top-3 features on the dataset of patient.info, the i2b2 dataset shows the second lowest F1-score ($\approx$61) i.e. $+1\%$ improvement w.r.t patient.info at the same feature setting. SVM shows the highest individual F1-score i.e. $\approx 93\%$ on patient.info with top-7 keywords followed by random forest $\approx 91\%$ which sets behind -2% in improvement.

Recall and precision as exhibits by naïve Bayes is uncanny. Almost in all experiments (i.e. except on patienet.info with top-5 and 7 features), it shows the highest value for precision $(90-97\%)$ along with the lowest value for recall $(52-67\%)$. SVM on dataset patient.info with top 7 keywords outperforms the results of all classifiers in achieving the desired high values in precision and recall.

Table IX provides the classification report of SVM on the dataset of patient.info. The values in the table correspond to the results of top-7 keywords, where SVM shows the highest performance. Similarly, for the dataset of i2b2, table X gives the classification report with decision tree as the classifying function. In both datasets, w.r.t these two tables (IX and X) we can see a the disease/class 'abdominal issues' secures the near-human predicting results. While, results for 'diabetics' (97%), 'heart' (98%), and 'fever' (96%) in patient.info (table IX), and 'diabetics' (92%) in i2b2 (table X) are also encouraging.

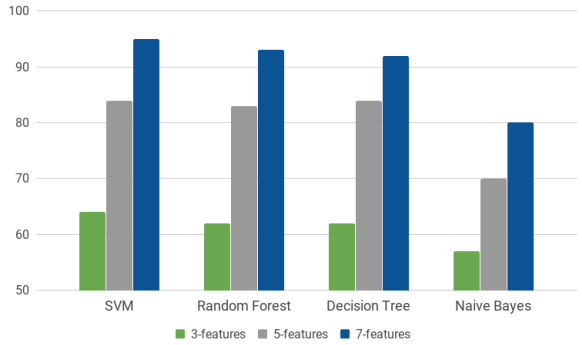
Collectively, on the basis of resulting accuracies, we can put forward that the naïve Bayes performs poor amongst the

TABLE VII. RESULTS OF DIFFERENT MATRICES ON DATASETS WITH TOP-5 FEATURES

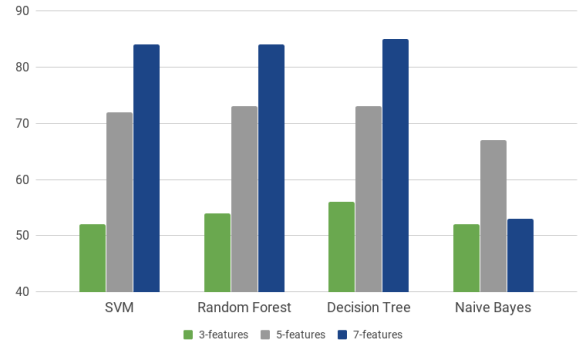
	patient.info (dataset A)			i2b2 (dataset B)		
Algorithm	Prec.	Rec.	F1-score	Prec.	Rec.	F1-score
SVM	85	84	83	88	73	74
Random Forest	85	81	81	86	73	74
Decision Tree	86	83	82	82	74	75
Naïve Bayes	75	74	73	90	67	72

TABLE VIII. RESULTS OF DIFFERENT MATRICES ON DATASETS WITH TOP-7 FEATURES

	patient.info (dataset A)			i2b2 (dataset B)		
Algorithm	Prec.	Rec.	F1-score	Prec.	Rec.	F1-score
SVM	95	92	93	88	84	84
Random Forest	93	91	91	88	84	84
Decision Tree	92	90	90	89	86	86
Naïve Bayes	80	79	79	97	53	64



(a) Confusion matrix of SVM on patient.info (dataset A) with top-7 features.



(b) Confusion matrix of Decision Tree on i2b2 (dataset B) with top-7 features.

Fig. 2. Confusion matrix of algorithms where they performed outstanding w.r.t patient.info and i2b2.

(a) Confusion matrix of SVM on patient.info (dataset A) with top-7 features.

(b) Confusion matrix of Decision Tree on i2b2 (dataset B) with top-7 features.

Fig. 3. Confusion matrix of algorithms where they performed outstanding w.r.t patient.info and i2b2.

TABLE IX. CLASSIFICATION REPORT OF SVM ON PATIENT.INFO (DATASET A) WITH TOP-7 FEATURES

Disease	Precision	Recall	F1-score
Abdominal Issues	1.00	1.00	1.00
Asthma	0.93	0.83	0.88
Diabetes	0.94	1.00	0.97
Heart	0.97	0.99	0.98
Hypertension	0.88	0.84	0.86
Fever	0.92	1.00	0.96

TABLE X. CLASSIFICATION REPORT OF DECISION TREE ON I2B2 (DATASET B) WITH TOP-7 FEATURES

Disease	Precision	Recall	F1-score
Abdominal Issues	0.99	1.00	0.99
Asthma	0.70	0.93	0.80
Diabetes	0.87	0.98	0.92
Heart	0.99	0.70	0.82
Hypertension	0.56	0.97	0.71
Fever	0.72	1.00	0.84

classifiers, while the SVM shows the highest figures followed by Random Forest and Decision Trees. The accuracy of the proposed system is also invigorating. Figure 2(b) shows that except the result of naïve Bayes on i2b2, every experiment shows the gradual increment in the accuracies as we move forward with the increment in keyword features. The highest accuracy yields on patient.info dataset are $\approx 95\%$ and on i2b2 is $\approx 85\%$ by SVM and decision tree respectively. These accuracies are yielded at the feature size of 7 keywords.

Figure 3 shows the errors and misclassification in the form of confusion matrices with the same experimental setting reported for tables IX and X. In figure 3, at x-axis there is the predicted and y-axis refers the actual classes. Since patient.info accounts hypertension as a sub-class in heart-related diseases therefore, we can give an empirical argument that the misclassifications are due to the nearly co-related diseases, like in figure 3(a) misclassifying 'asthma' as 'heart', 'hypertension', and 'fever', and in figure 3(b) misclassifying 'heart' as 'hypertension', 'asthma' and 'fever'. In the entire experimental suit, the highest misclassification is seen in i2b2, specifically 'heart' as 'hypertension'.

VI. CONCLUSION

In the medical field mostly the problems need their solutions for the betterment of the society at a broader level. The natural language processing can help many things related to the text. The patient descriptions in our local context are written in the form of textual format. In this study, we have developed a

solution for the medical practitioners and the 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 suggest the preventions based on the information given in the 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 got good results with small datasets. Also, we have calculated the results of multiple keywords and key phrases. The results shows SVM as the classifying champion amongst naïve Bayes, Random Forest, and Decision Tree algorithms.

VII. Future Work

The real-time and problem related local context based dataset was the bigger challenge. In future, this work can be improved with more optimal results by utilising the Named Entity Recognition (NER) with word embedding techniques and deep learning algorithms. Also, we see this work as an extension towards the chatbot form with the large dataset.

References

[1] B. Djulbegovic and G. H. Guyatt, "Progress in evidence-based medicine: a quarter century on," *The Lancet*, vol. 390, no. 10092, pp. 415–423, 2017.

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

[3] A. Névéol, H. Dalianis, S. Velupillai, G. Savova, and P. Zweigenbaum, "Clinical natural language processing in languages other than english: opportunities and challenges," *Journal of biomedical semantics*, vol. 9, no. 1, p. 12, 2018.

[4] X. Guo, T. Klinger, C. Rosenbaum, J. P. Bigus, M. Campbell, B. Kawas, K. Talamadupula, G. Tesauro, and S. Singh, "Learning to query, reason, and answer questions on ambiguous texts," 2016.

[5] D. Mollá, M. E. Santiago-Martínez, A. Sarker, and C. Paris, "A corpus for research in text processing for evidence based medicine," *Language Resources and Evaluation*, vol. 50, no. 4, pp. 705–727, 2016.

[6] M. Conway, S. Doan, A. Kawazoe, and N. Collier, "Classifying disease outbreak reports using n-grams and semantic features," *International journal of medical informatics*, vol. 78, no. 12, pp. e47–e58, 2009.

[7] İ. Dönmez and E. Adali, "Extracting phrase-content pairs for turkish sentences," in *Application of Information and Communication Technologies (AICT), 2015 9th International Conference on*. IEEE, 2015, pp. 128–132.

[8] G. A. Miller, "Wordnet: a lexical database for english," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[9] A. Chandurkar and A. Bansal, "Information retrieval from a structured knowledgebase," in *Semantic Computing (ICSC), 2017 IEEE 11th International Conference on*. IEEE, 2017, pp. 407–412.

[10] S. N. Kim, D. Martinez, L. Cavedon, and L. Yencken, "Automatic classification of sentences to support evidence based medicine," in *BMC bioinformatics*, vol. 12, no. 2. BioMed Central, 2011, p. S5.

[11] A. Sarker, D. Mollá, and C. Paris, "Query-oriented evidence extraction to support evidence-based medicine practice," *Journal of biomedical informatics*, vol. 59, pp. 169–184, 2016.

[12] H. Yu and Y.-g. Cao, "Automatically extracting information needs from ad hoc clinical questions," in *AMIA annual symposium proceedings*, vol. 2008. American Medical Informatics Association, 2008, p. 96.

[13] W. Pratt and L. Fagan, "The usefulness of dynamically categorizing search results," *Journal of the American Medical Informatics Association*, vol. 7, no. 6, pp. 605–617, 2000.

[14] Y. Cao, F. Liu, P. Simpson, L. Antieau, A. Bennett, J. J. Cimino, J. Ely, and H. Yu, "Askhermes: An online question answering system for complex clinical questions," *Journal of biomedical informatics*, vol. 44, no. 2, pp. 277–288, 2011.

[15] S. Rose, D. Engel, N. Cramer, and W. Cowley, "Automatic keyword extraction from individual documents," *Text Mining: Applications and Theory*, pp. 1–20, 2010.

[16] S. Bird and E. Loper, "Nltk: the natural language toolkit," in *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*. Association for Computational Linguistics, 2004, p. 31.

[17] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[18] J. R. Quinlan, "Induction of decision trees," *Machine learning*, vol. 1, no. 1, pp. 81–106, 1986.

[19] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[20] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[21] P. P. Limited. (2018) Symptom checker, health information and medicines guide — patient. Patient Platform Limited. Accessed: 2018-06-27. [Online]. Available: https://patient.info/

[22] i2b2. (2018) i2b2: Informatics for integrating biology & the bedside. Partners Healthcare. Accessed: 2018-06-27. [Online]. Available: https://www.i2b2.org/NLP/DataSets/

[23] J. Patrick and M. Li, "High accuracy information extraction of medication information from clinical notes: 2009 i2b2 medication extraction challenge," *Journal of the American Medical Informatics Association*, vol. 17, no. 5, pp. 524–527, 2010.