Performance Analysis of Learning Models on Medical Documents

Vanitha Guda[1], Manish Golla[2], Akhilesh Datta[3]
[1]*Assistant Professor, CSE Department, Chaitanya Bharathi Institute of Technology (A),Hyderabad.*
[2,3]*BE Computer Science Engineering, CBIT(A),Hyderabad*

Abstract- **With the exponential growth of online text, Text Classification domain becomes the major field of Natural language Processing and Machine learning. In this context Medical Document Classification is one of the popular research problem to analyze the high dimensionality features of medical data. Our Study considered various learning models and their performances over the medical documents and we considered OSUMED is one of the popular datasets containing MEDLINE documents as multi-labelled documents. Choosing a high accuracy classifier for text classification is still a challenging task for many of the practitioners. Our work aims to find the efficiency in classifiers and comparing the accuracy in classifying medical documents with well-known classifiers Naïve Bayes, Decision Tree, Support Vector Machine (Linear) and Stochastic Gradient Descent (SGDC). The performance of a feature selection method namely Univariate Feature Selection is analyzed using pattern classifiers namely Naïve Bayes, Decision Tree, Support Vector Machine (Linear) and SGDC and the obtained experimental results shows that the combination of Univariate Feature Selector and Support Vector Machines classifier gives more accurate results in most cases than the others.**

Index Terms- **Classifier's Accuracy, Document classification, Feature Selection, Learning Models, Medical Documents, Text Classification.**

I. INTRODUCTION

With the rapid growth in usage of web sources, Internet technology leads to proportional increment to the generation of electronic documents in this context the concept of automatic text categorization and classification got significant importance. Automatic text classification is an approach which assigns the electronic documents to the referred and appropriate classes based on the content [1]. Text classification is the process to solve different problems like filtering of classification of web pages [2], author identification [3], spam e-mails [4], and classification of medical text documents [5][6][7].

In the research field of Text classification medical documents classification is the specific task. Most of the researches relate the medical abstracts from the MEDLINE database [8], it is a bibliographic database containing nearly 21 million documents, about 5600 medical journals, and it consists of medical abstracts in english those are assigned to some categories namely medical subject headings (MeSH). Osumed is the most used dataset for automatic classification of MEDLINE documents, it is a multi-label in structure and contains medical abstracts in English for 23 types of diseases.

Several existing works carried out in the field of medical domain, but most of the existing works and the studies are about the usage of medical words, phrases, and their combinations as features for the document classification. The obtained results are explaining that using combination of words and phrased as features gives slightly better classification performances than the others. In another work the study about multi-label classification performance based on associative classifier is examined on medical articles [9], in another work, HMM - hidden Markov models are used for classification [10]. One of the recent works, an approach using support vector machines and latent semantic indexing is applied to some datasets including the ones consisting of medical abstracts [11].

Classification is one example of pattern recognition. In Daily routines hospital databases generates huge amounts of data and most of the researchers in this field evaluate their classification methodologies on medical documents retrieved from MEDLINE database. Extraction of useful information from online is a challenge because most of the documents

are not in structured form. With a proper study of the existing work, we noticed that the highest accuracy of the learning models studies is 72% for Distinguished Feature Selection and Bayesian Network Combination. The researcher did not take into consideration other classifiers such as SVM and Naïve Bayes. Choosing a high accuracy classifier for text classification is still a challenging task for many of the practitioners.

In our present work, classifying [12] medical documents with the well-known classifiers Naïve Bayes, Decision Tree(DT), Support Vector Machine(SVM-Linear), Stochastic Gradient Descent(SGDC),and comparing the efficiency and accuracy of these classifiers. Remaining sections of the paper organized as follows section -2 presents System Architecture, which is the core part of the work. Section-3 explains the implementation of the algorithms used in building of the architecture. Section- 4 is the results and discussions through a series of screenshots of obtained from the executed results; finally section-5 is the conclusions.

II. SYSTEM ARCHITECTURE

Extracting the required features from the given sources of text and using the generated features for further processing is our major task. The architecture consists of main modules are Feature extraction module, Feature Selection module, Pattern Classifiers and measuring the accuracies of the classifiers in Performance analysis module.

Figure 2.1 depicts the architecture of the system, as an input source it considers medical documents or medical database in this work we have taken Osumed database which contains thousands of documents. The documents in the database are not processed and most of them are redundant also. To avoid the duplicates and to obtain the better accuracy first we performed removal of `multi label documents which can eliminate the duplicated data.

The next module is Feature Extraction it contains sub modules of Pre-Processing and TF-IDF representations. Pre-processing deals with Stop Word Removal, Parts Of Speech (POS) tagging, Stemming and TF-IDF representation of the documents. Next module Feature Selection selects the required features by applying several algorithms on the extracted features after that by using pattern classifier module classifying the features. The experimental results shows that the most successful setting is the combination of Univariate with Support Vector machine classifier. The modules explained in detail in next sections

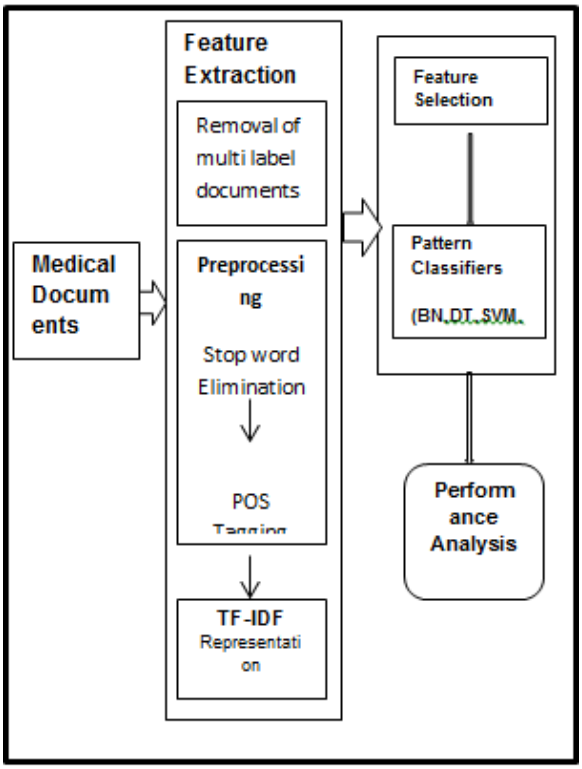

Figure 2.1: System Architecture with Modules

III. IMPLEMENTATION

The major modules of the architecture are Feature Extraction, Feature Selection, Pattern classifiers and Performance analysis. Each module is individual in operation but specific in performance.

3.1. Feature Extraction- In Pattern recognition and Image processing field feature extraction starts from an initial set of measured data and builds derived values or features intended to be informative and non-redundant. Facilitating the subsequent learning and generalization steps for better human interpretations feature extraction is relates to dimensionality reduction. The following are the steps in feature extraction-

A. Removal of Multi-Labelled Documents
B. Stop Word Removal
C.POS Tagging
D. Stemming
E.TF-IDF Representation

A) Removal of Multi-Labelled Documents: In the first step of feature extraction, documents belonging to multi class are removed which results to single labelled documents for further processing.
Procedure for RemMutliLabel (Dataset)
Begin
Consider all the multi-labelled documents in one directory.
Next consider file names of a particular root directory and copy them into a buffer array x
Compare the names of the files in array x with all the successive files of directory and retain only those names in x,which don't have any duplicates in the successive directories
Copy back the file names in array x into the Original folder from which it was constructed
Repeat above steps for all the documents of that directory or other directory
End.

B) Stop Word Removal: Removing unnecessary words like a, an, the… etc. from text for easy processing. Split all the text into word segments and remove the words matched with the stop words which are already stored in an array.
Procedure for StopWordRem(Documents)
Begin
Consider a variable which stores all the stop words of 'english' language, punctuations marks and other meta characters. Compare the given documents with the text file with the above variable and remove all the entities which matches with the entities present in the variable.
Now write the file to a different new respective Documents.
End

C). POS Tagging: For a given word, POS tagging means to decide which morph syntactic class it belongs to and assigning the word to that class.
Procedure for POS Tagging(Documents)
Begin
Apply word tokenizer to a file and store it in a variable.
Apply POS Tagging to the words obtained from step1 and store the tags generated in a list.
Consider a new list with the name of tags which are essential for Medical Documents Dataset.
Compare the new list with the list obtained after step1 and store only the words whose tags match with new list in a separate file
Store the files in the new respective Documents.
End
•Nltk module is used in this step, first all the words are tokenized into the variable text_pos by calling the nltk.word_tokenize() method and now the variable text_pos is considered for POS tagging through the method nltk.pos_tag(text_pos) and stored in a new_list1.This new_list1 is compared with the list which consists of relevant tags and those words whose tags matches with the relevant tags are kept ,rest are removed

D) Stemming: Stemming is also one of the preprocessing step, depends on the word stems root word can easily analyzed and stemming also reduces the count of a word occurrence in the directory with its root word.
Procedure Stemming (Documents):
Begin
Open the file from document and apply Porter Stemming Algorithm to all the words in a file
Write the generated words obtained from the previous step into a new file in new respective documents
End
•Porter Stemming algorithm is used for stemming and considered under the variable ps, through ps= PorterStemmer(),all the files are recursively run in a loop and for each and every file Porter Stemming Algorithm is applied through ps.stem() and the resultant files are stored in the respective new directory.

E) TF-IDF Representation: It is the third step of feature extraction, the documents are represented in TF-IDF form which assigns a TF-IDF value for each feature in the document. This can be calculated as product of TF*IDF of each word, where
TF(t)= (Number of times term t appears in a document) / (Total number of terms in the document).
IDF(t)= log_e(Total number of documents / Number of documents with term t in it).
This representation is in the form of sparse matrix where each row represents a feature vector
Begin

Calculate TF: Term Frequency, which measures how frequently a term occurs in a document.
TF(t) = (Number of times term t appears in a document) / (Total number of terms in the document)
Calculate IDF: Inverse Document Frequency, which measures how important a term is.
IDF(t) = log_e(Total number of documents / Number of documents with term t in it)
End

3.2. Feature Selection- Feature selection in machine learning termed as variable selection, variable subset selection or attribute selection and it is the process of selecting a subset of relevant features or variables or predictors for the model construction. Feature selection techniques mainly fall into three categories: filters, wrappers, and embedded methods. Filters are the techniques and computationally these are fast, but usually do not take feature dependencies into consideration [1]. Filter-based methods are widely preferred especially for text classification domain. Feature selection techniques are used majorly for four reasons:

- For the users and researches able to make the interpretations in easy way from the simplified models
- shorter training times
- to avoid the curse of dimensionality
- enhanced generalization by reducing over-fitting

In this work for Feature Selection Method called Univariate Feature Selection Algorithm for Univariate Feature Selection for variant of features. SelectKBest by chi2 statistical test is applied to Sparse matrix of TF-IDF through the methods of X_new=SelectKBest(chi2,k=kk).fit_transform(X_train_tfidf,med.target), where kk represents the no of features to select and transformed using fit transform() functions which takes parameters of data to train and target variables.

3.3 Pattern Classifiers- In our work there are several existing classifiers like Bayesian Networks and C4.5 decision tree classifiers and support vector machines and Stochastic Gradient Descent are used. These algorithms are explained in the subsection below.

A. Bayesian Networks (BN)

Bayesian Network (BN) is the methods which is used to denote modelling and state transitions [13]. BN is often used for modelling discrete and continuous variables of multinomial data. These networks encrypt the relationships between variables in the modelled data. In BN, the nodes are interconnected by arrows to indicate the direction of engagement with each other.

B.C 4.5: Decision Tree (DT)
The main purpose of the decision tree algorithms is to split the feature space into unique regions corresponding to the classes .An unknown feature vector is assigned to a class via a sequence of Yes/No decisions along a path of nodes of a decision tree. C4.5 is an algorithm used to generate a decision tree and it is known as one of the successful decision tree classification algorithms.

C. Naïve Bayes (Features) :
Begin Consider D: Set of tuples Each tuple is an 'n' dimensional attribute vector X : (x1,x2,x3,…. xn)
Let there be 'm' Classes : C1,C2,C3…Cm Naïve Bayes classifier predicts X belongs to Class Ci if
P (Ci/X) > P(Cj/X) for 1<= j <= m , j <> i
Maximum Posteriori Hypothesis P(Ci/X) = P(X/Ci) P(Ci) / P(X)
Maximize P(X/Ci) P(Ci) as P(X) is constant Naïve Assumption of "class conditional independence"
$P(X/Ci)=\pi P(/xk/Ci)$
P(X/Ci) = P(x1/Ci) * P(x2/Ci) *…* P(xn/ Ci)
End

D) Support Vector Machines: SVM(Features)
Begin
If the training data is separable, then select two hyper planes in a way that they separate the data.
Calculate the distance between these two hyper-planes by applying simple geometry.
Measure distance directly by 2/‖a‖ quantity. To increase the distance, have to reduce‖a‖
Use standard quadratic programming techniques and programs to solve the problem using rimal form use thedual form to write classification rules as an unconstrained system.
By doing this you get hyperplane with greatest possible margin. Then represent classification process as a function of support vector machines [14].

Represent data points and hyperplanes in the same coordinate system
End
•Support Vector Machine classifier is applied through the function clf = LinearSVC(random_state=0) and text_clf_svm=clf.fit(X_train,y_train) where the fit functions takes the training data for features and target variable respectively.Then the classifier is used to predict the target variables by giving some test data through the function predicted_svm=text_clf_svm.predict(X_test).

E) Stochastic Gradient Descent (SGDC)
Algorithm SGDC(Features)
Begin
Updates the parameters θ of the objective J(θ) as, $\theta=\theta-\alpha\nabla\theta E[J(\theta)]$ expectation in the above equation is approximated by evaluating the cost and gradient over the full training set.
Computes the gradient of the parameters using only a single or a few training examples
The new update is given by, $\theta=\theta-\alpha\nabla\theta J(\theta;x(i),y(i))$ with a pair (x(i),y(i)) from the training set.
Each parameter update in SGD is computed w.r.t a few training examples or a mini-batch as opposed to a single example.
Choosing the proper learning rate(α) and schedule
Randomly shuffle the data prior to each epoch of training
End

3.4 Performance Analysis - The Performance of the Classifiers are evaluated on the basis of performance metric known as F1 score. It considers both the precision P and the recall R of the test to compute the score: "P" is the number of correct positive results divided by the number of all positive results which are returned by the classifier and the recall " R" is number of correct positive results which are divided by the number of all relevant samples.

- Precision: It is the ratio of tp/(tp+fp) where, tp is the number of true positives and fp the number of false positives. The ability of the classifier not to label as positive which is negative in terms.
- Recall : Recall is the ratio tp/(tp+fn) where, tp is the number of true positives and fn the number of false negatives. The recall is the thing to find all the positive samples.
- F1 score : It is is the harmonic average of the precision and recall, where an F1 score reaches its best value at 1 (perfect precision and recall) and worst at 0 and F1 = 2 * (precision * recall) / (precision + recall).

IV. EXPERIMENTAL RESULTS

This section provides the investigations and observations of the experiment and the results to measure the performance of feature selection and classifiers. For the analysis, the combinations of feature selection methods with BN, DT, SVM and SGDC classifiers were analyzed in order to determine the best combination of the classifiers performance over the datasets. In the following subsections, the utilized datasets and success measures are briefly described. Then, the experimental results are presented.

A. Datasets In our work we used MEDLINE documents, which are a well-known Osumed dataset, it consists of medical abstracts collected with 23 cardiovascular disease categories. Mainly it deals with single-label text classification the documents belonging to multiple categories are eliminated. Only 10 classes are used for classification in order to make the class distribution same with the second dataset. The documents having multiple categories are removed from this dataset because of concerning single-label classification of medical documents. MEDLINE documents only originated from medical journals in Turkey rather than originating from different locations. In the experiments, seventy percent of documents in each class was used training and the rest was used for testing.

B. Execution in Steps

1. First step removal of the multi-labelled documents are removed from the data set.
2. Second perform Stop Word Removal and POS Tagging
3. Perform Stemming
4. Perform TF-IDF representation to construct Sparse Matrix and apply classifiers

The figure-2 shows the TF-IDF representation of the document term matrix that we have achieved using all the documents after pre-processing. The TF-IDF is the representation in the form of sparse matrix

whose dimension is 18302 x 33059, which represents there are 18302 documents(rows) and 33059 words(columns).

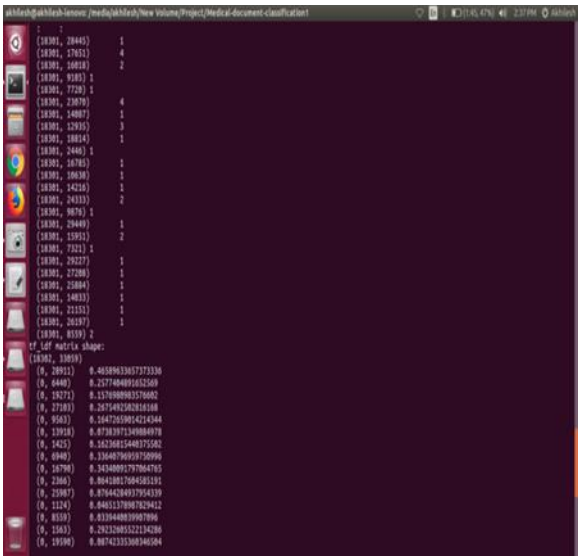

Fig-2: Sparse Matrix Representation

C. Performance Analysis of Classifiers

Table-1: Accuracies of Classifiers

Classifiers with Univariate features	Number of features	Precision	Recall	F1-score
Univariate + Naïve Bayes	500	0.63	0.46	0.41
	1500	0.64	0.54	0.50
	2000	0.65	0.54	0.50
	3000	0.65	0.54	0.50
Univariate + SVM	500	0.69	0.67	0.68
	1500	0.74	0.74	0.74
	2000	0.75	0.75	0.75
	3000	0.76	0.76	0.76
Univariate + SGDC	500	0.65	0.67	0.65
	1500	0.72	0.72	0.70
	2000	0.73	0.74	0.72
	3000	0.74	0.75	0.73
Univariate + DT	500	0.57	0.56	0.56
	1500	0.58	0.58	0.58
	2000	0.58	0.58	0.58
	3000	0.56	0.56	0.56

In Table-1 the first column represents classifiers of Univariate with Naive Bayes, SVM,DT, and SGDC, second column.is number of features which can be used for train the model, last three are the accuracy (score) of the learning model precision, recall, f1-score, support of documents in average. The highest accuracy measures for univariate and SVM with highest precision i.e, 76 this combination can be used for highest feature classification.

V. CONCLUSION

In this Work the performance of four widely known classifiers Multinomial Naive Bayes, Decision Tree ,Support Vector Machines and Stochastic Gradient Search are extensively analyzed using feature selection methods: Univariate Method. By Comparing theire performance we observed that the Learning Model i.e. combination of Univariate Feature Selection and Support Vector Machine gives the highest accuracy of 76.19 for 3000 features. POS Tagging plays a significant role in pre-processing step, in our case words were reduced to a great a extent after applying this technique. As a future work, a new dataset containing Turkish versions of the documents in the self-constructed dataset may be compiled and classification performances of these two datasets having same documents in different languages can be extensively analyzed.

REFERENCES

[1] Uysal, A. K., & Gunal, S. A novel probabilistic feature selection method for text classification. Knowledge-Based Systems, 36, 226-235,2012.

[2] Özel, S. A. A Web page classification system based on a genetic algorithm using tagged-terms as features. Expert Systems with Applications,38(4), 3407-3415, 2011.

[3] Zhang, C., Wu, X., Niu, Z., & Ding, W. Authorship identification from unstructured texts. Knowledge-Based Systems, 2014.

[4] Idris, I., Selamat, A., Nguyen, N. T., Omatu, S., Krejcar, O., Kuca, K., & Penhaker, M. A combined negative selection algorithm–particle swarm optimization for an email spam detection system. Engineering Applications of Artificial Intelligence, 39, 33-44, 2015.

[5] Garla, V., Taylor, C., & Brandt, C. Semi-supervised clinical text classification with Laplacian SVMs: An application to cancer case management. Journal of biomedical informatics, 46(5), 869-875, 2013.

[6] Yetisgen-Yildiz, M., & Pratt, W. The effect of feature representation on MEDLINE document classification. In AMIA annual symposium proceedingsAmerican Medical Informatics Association, (Vol. 2005, p. 849),2005.
[7] Yepes, A. J. J., Plaza, L., Carrillo-de-Albornoz, J., Mork, J. G., & Aronson, A. R. Feature engineering for MEDLINE citation categorization with MeSH.BMC bioinformatics, 16(1), and 1, 2015.
[8] MEDLINEhttp://www.nlm.nih.gov/databases/databases_medline.html] Accessed 2015
[9] Rak, R., Kurgan, L. A., & Reformat, M. Multilabel associative classification categorization of MEDLINE articles into MeSH keywords. IEEE engineering in medicine and biology magazine, 26(2), 47, (2007)..
[10] Yi, K., & Beheshti, J. (2008). A hidden Markov model-based text classification of medical documents. Journal of Information Science
[11] Uysal, A. K., & Gunal, S. Text classification using genetic algorithm oriented latent semantic features. Expert Systems with Applications, 41(13), 5938-59, 2014.
[12] Pubmed [http://www.ncbi.nlm.nih.gov/pubmed]. Accessed 2015.
[13] Ian H Witten and Eibe Frank, Data Mining: Practical Machine Learning Tools and Techniques, Jim Gray, Ed. San Fransisco: Morgan Kaufmann Publishers, 2005.
[14] Garla, V., Taylor, C., & Brandt, C. Semi-supervised clinical text classification with Laplacian SVMs: An application to cancer case management. Journal of biomedical informatics, 46(5), 869-875,2013.