Named Entity Recognition for the Medical Domain: A Machine Learning Approach

Abhijeet Anand
Dept. of Information Science
PESIT
Bangalore, India
abhijeet.anand804@gmail.com

Akshat Maheshwari
Dept. of Information Science
PESIT
Bangalore, India
akshatmaheshwari1995@gmail.com

Himanshu Singh
Dept. of Information Science
PESIT
Bangalore, India
kishu0495@gmail.com

Dr. Shylaja SS
Head, Dept. of Information Science
PESIT
Bangalore, India
shylaja.sharath@pes.edu

Abstract—**Medical Entity Recognition is a crucial step towards efficient medical text analysis. In this paper, we presented and compared two machine learning algorithms (namely KNN and SVM) that can be used for classifying a text containing medical data into an appropriate predefined category. To apply these algorithms the first task that was needed to be performed was appropriate feature selection. For this, we incorporated NLP techniques, such as chunking and stemming, on the text provided. The above approaches were tested on a standard corpus of medical texts. The obtained results show that the hybrid approach based on both machine learning and NLP (for feature selection) techniques performed well for medical text categorization.**

Keywords—NLP, KNN, SVM, Supervised Machine Learning Algorithms,Feature selection,Chunking, Stemming, Dimensionality reduction,Named Entity Recognition

I. INTRODUCTION

With the rapid growth of online information, text categorization has become one of the key techniques for handling and organizing text data. Text categorization techniques are used to classify news stories, to find interesting information on the WWW, and to guide a user's search through hypertext. Since building text classifiers by hand is difficult and time consuming, it is advantageous to learn classifiers from examples.

A growing number of statistical classification methods and machine learning techniques have been applied to text categorization in recent years including multivariate regression models, nearest neighbour classification, Baye's probabilistic approaches, decision trees, neural networks and inductive learning algorithms.

A major characteristic, or difficulty, of text categorization problems is the high dimensionality of the feature space. The native feature space consists of the unique terms (words or phrases) that occur in the document which can be tens or hundreds of thousands of terms for even a moderate-sized text collection. This is prohibitively high for many learning algorithms. Therefore, it is highly desirable to reduce the native space without sacrificing categorization accuracy. It is also desirable to achieve such a goal automatically, i.e., no manual definition or construction of features is to be required.

Automatic feature selection methods include the removal of non-informative terms according to corpus statistics, and the construction of new features which combine lower level features (i.e., terms) into higher level orthogonal dimension.

In this paper we have presented two machine learning techniques namely, K-Nearest Neighbour (KNN) and Support Vector Machine (SVM) for medical text classification. These algorithms were applied on a custom corpora and the accuracy was observed on the selected features for each classification category.

II. METHODOLOGY

A. Data Set / Corpora

We used Ohsumed collection [[2]Yang] as our custom corpora or data set for training and testing the learning algorithms.

The Ohsumed collection includes 50,216 medical abstracts divided into 23 categories. The first 30,000 documents categorized into 22 categories are used in our experiments.

We used 80% of the text files as our training data and remaining 20% as the testing data. There were 16832 unique features obtained from all the categories.

B. Processing

The data present in the collection was in the form of raw text for which some processing had to be done to apply learning algorithms to them and thus, classifying them. The appropriate features were extracted by applying Natural Language Processing techniques, such as chunking and stemming of the words along with POS tagging. After extracting the most appropriate features from all the categories, each text file was represented as a feature vector and was given as the input to the Supervised learning algorithms (namely KNN and SVM) for the classification purpose.

C. Analysis

After the feature extraction and the implementation of the algorithms, the performance of the respective algorithms was analyzed on the basis of the percentage of correct predictions.

After finding the accuracy of both the algorithms on each category of the Ohsumed collection, it was found that SVM outperforms KNN algorithm on a number of categories and hence overall SVM proved to be a better Supervised Learning Algorithm as compared to KNN in medical text classification.

III. Algorithm and implementation

A. *Feature Selection [[7]Sam Scott]*

Feature Selection (FS), a pre-processing technique, is used to identify the significant attributes, which play a dominant role in the task of classification. This leads to dimension reduction. By applying different approaches, features can be reduced. The reduced feature set improves the accuracy of the classification as compared to applying the classification on the original data set. The overall procedure includes the following steps:

a. Pre-processing of data which is in any format

b.Selection of attributes using feature selection for dimension reduction.

c. Data set with reduced set of attributes given as input to the classifier.

The type of representation that dominates the text classification literature is known as the "bag of words". For most bag of words representations, each feature corresponds to a single word found in the training corpus, usually with case and punctuations removed. Infrequent and frequent words are separated, with words in the stop words' corpus removed. The chief advantage of removing infrequent words is reduction in feature size.

In order to make the features more statistically independent, we removed the suffixes from words using a stemming algorithm provided in NLTK (a standard toolkit for NLP related facilities) as Porter Stemmer.

Noun Phrase Extraction: [[1]Alessandro Moschitti]

It can be observed that sometimes a collection of words as a feature rather than just word by word, increases the expressiveness of the feature. For example, the number 23 or the word 'weeks' have no sense if they are taken alone. Instead the complex nominal '23_weeks' evokes a recurrent period of time of pregnancy.

We performed Noun Phrase Extraction as a part of our feature selection through chunking with POS tagging (both provided by NLTK).

Document Frequency Thresholding (DF):[[2]Yiming Yang]

Document Frequency is the number of documents in which the term occurs. We computed the frequency of each term in a category in the training corpus and removed those terms from the feature space whose document frequency was less than some predefined threshold.

The basic assumption is that rare terms are either non-informative for category prediction, or not influential in global performance.

DF thresholding is the simplest technique for vocabulary reduction. It easily scales very large corpora with a computational complexity, approximately linear in the number of training documents.

B. *Support Vector Machine (SVM) Approach [[3]Peter Harrington]*

Support Vector Machines are supervised learning methods with associated learning algorithms that analyze data and recognize patterns used for classification and regression analysis. SVM training algorithm builds a model that assigns new examples into one category or other, making it a non-probabilistic binary linear classifier. Construction of a hyper plane in a finite or an infinite dimensional space is used for classification and regression. A good separation is achieved by the hyper plane that has the largest distance to the nearest training data point of any class namely the functional-margin.

Some dependencies for applying SVM are Matplotlib for the eventual data visualization; the SVM Scikit learn classifier [4] SVC (Support Vector Classifier) accepts tuples in the order (1), (kernel, constant(C)) where kernel is a similarity function and constant C is amount of proper classification which are chosen as linear and 1.0 respectively.

clf = svm.SVC (kernel='linear', C=1.0) (1)

For this experiment, we have applied SVM algorithm to each of the 22 categories and have followed 'one versus rest' approach for linear SVM.

In the 'one versus rest' approach, we created feature vector for all files in all categories for each of the 22 categories where each feature vector consists of two components, the first being the number of features matching with the unique features provided for that category and the other being the number of features not matching with the unique features for that category.

We trained the classifier using SVM classifier which is present as a library function in the Scikit-learn package, by providing the feature vector for each training file as numerical co-ordinates in an array using Numpy (2) and also passing a vector containing only zeroes and ones where the file corresponding to one representing that it belongs to this particular category whereas zero depicting exactly the reverse. The SVM classifier generates the hyper plane using the *fit()* function (3) and helps to predict the category of the testing data by finding the position of the co-ordinate on one side of the hyper-plane (here line) on that plane by classifying them based on the features.

X = numpy.array(training_data) (2)

clf.fit(X, y_train) (3)

The hyper plane is presented as in Figure1. This figure depicts how the feature vectors get generated as numerical values and are separated by the hyper plane for an efficient classification process by the SVM classifier.

The red marked points represent the feature vector of those files which are present in that particular category and blue points represent the feature vector of those files which are not present in that category.

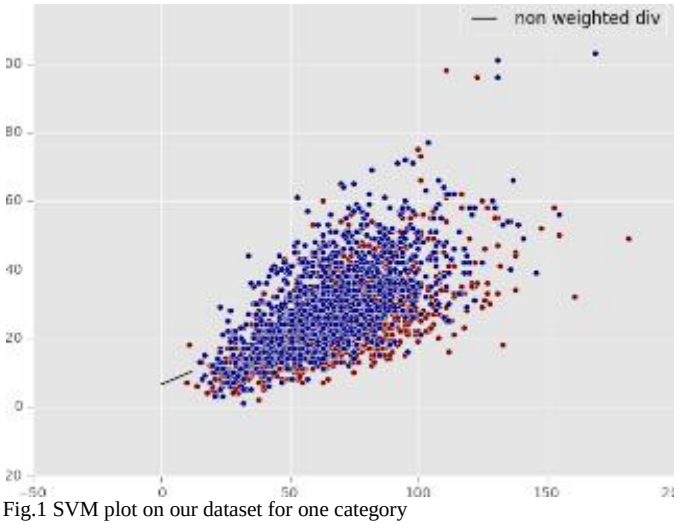

Fig.1 SVM plot on our dataset for one category

C. K-Nearest Neighbor(KNN) Approach[[3]Peter Harrington]

The K-Nearest Neighbor algorithm is yet another supervised machine learning algorithm that we have used in our experiment for text classification.

When classifying with the help of KNN, the output is a class membership. An object is classified by majority vote of its neighbors, with the object being assigned the class which is most common among its K nearest neighbors (K is a positive integer typically small).

KNN classifiers were found to show very good performance on text categorization tasks [[2]Yang].

To apply the KNN algorithm to our data set, the first step that we did was to convert all training files in all the categories to feature vectors, where the components of a feature vector are the unique features obtained from all the categories. Therefore, each file was converted into a vector containing only zeroes and ones where ones represent that the file contains the corresponding feature and zeroes represent that the file doesn't contains the corresponding feature.

The vectors thus obtained was given as input to the Nearest Centroid (a standard library function in the Scikit learn package), in the form of a matrix, along with a vector representing the class label of each training file in the array, for training the classifier. The Nearest Centroid classifier uses the *fit()* method to train the classifier(1), and uses the *predict()* method to predict the category for testing the files (2).

clf.fit(feature_knn, cat_knn) (1)

prediction_vector = clf.predict(file_vector_each_category) (2)

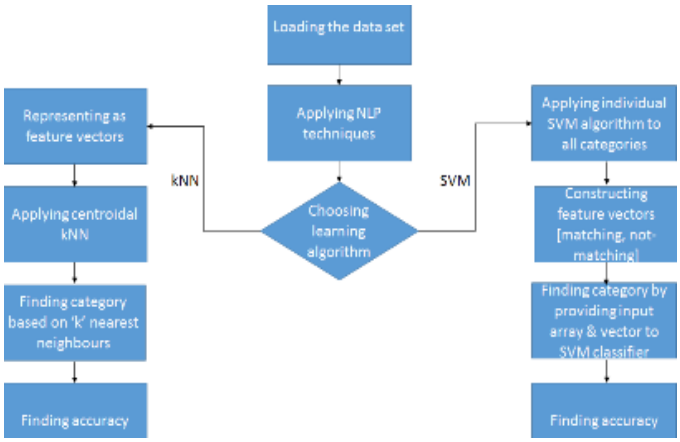

Fig.2 Data flow diagram

IV. RESULTS AND DISCUSSION

The result of our project was to classify a given medical text into one of the predefined categories. The two learning algorithms were applied and accuracy obtained clearly shows that SVM outperforms KNN in almost every category, making SVM a better choice for text categorization.

The accuracy depends upon the feature selection and feature vector formation for both the algorithms as they are the base for the classifiers. Figure3 depicts the comparative plot of the accuracy of the two supervised machine learning algorithms for some of the disease categories.

TABLE I. ACCURACY RESULTS OF SUPERVISED LEARNING ALGORITHMS

Disease Category	**Accuracy Percentages**	
	SVM	***KNN***
Virus	84.62	80.17
Musculoskeletal	65.61	53.37
Stomatognathic	68.57	63.62
Nervous System	69.38	50.58
Eye	83.21	68.83

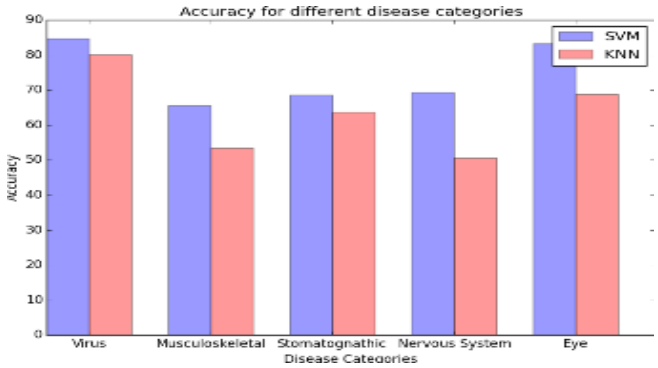

Fig 3.A comparative bar plot of different disease categories of the two supervised learning algorithms

Feature extraction, processing time, reliability of the results and data set type are the factors which can vary the results of any of the suggested algorithms. The only limitation of the project being that the supervised learning algorithms takes a fair amount of time to train and test the data which pulls the performance of the system to a lower level.

V. CONCLUSION AND FUTURE WORK

In this paper, we incorporate the supervised learning algorithms of machine learning along with pre-processing with NLP tools for the category classification of a medical text. Firstly, custom corpora was used for training and testing purposes. Then, for the two algorithms prescribed, the feature extraction took place. Depending upon the classifier, the algorithms make use of different feature vectors, i.e. the feature vectors for KNN were different from those for SVM. KNN works on the instance based learning approach while SVM works for category detection through the hyper planes concept. The results and experiments tell about the efficiency of the proposed methods and their comparison for various categories of the collection.

Future work includes a better feature extraction for all the proposed methods so as to increase the accuracy measure of the medical text classification. We also plan to apply some other machine learning algorithms such as Neural Networks and compare the accuracy with the current obtained results.

In addition, we plan to have a doctor recommendation system which will recommend the appropriate doctor based on the medical history of a patient.

VI. ACKNOWLEDGMENT

We would like to thank Dr. Shylaja SS, Professor & Head of Department-ISE, PES Institute of Technology, Bangalore, for her immense encouragement and guidance throughout the project. We would also like to thank our Information Science Dept. for their kind support.

VII. REFRENCES

[1] Alessandro Moschitti
"Natural Language Processing and Automatic Text Categorization: A study of the reciprocal beneficial interactions"
[2] Yiming Yang, Jan O.Pedersen
"A Comparative Study on Feature Selection in Text Categorization".
[3] "Machine Learning in Action" by Peter Harrington.
[4] http://pythonprogramming.net/linear-svc-example-scikit-learn-svm-python/
[5] Asma Ben Abacha, Pierre Zweigenbaum.
"Medical Entity Recognition: A comparison of Semantic and Statistical Methods"
[6] Thorsten Joachims
"Text Categorization with Support Vector Machines: Learning with Many Relevant Features."
[7] Sam Scott, Stan Matwin
"Feature Engineering for text Classification".
[8] Prof. K. Rajeswari, Dr. V. Vaithiyanathan and Shailaja V. Pede
"Feature Selection for Classification in Medical Data Mining"