International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 4 Issue 02, February-2015

Comparison of Text Classification Algorithms

M. Trivedi
Dept. of Computer Engineering
D. J. Sanghvi College of Engineering
Mumbai, India

S. Sharma
Dept. of Computer Engineering
D. J. Sanghvi College of Engineering
Mumbai, India

N. Soni
Dept. of Computer Engineering
D. J. Sanghvi College of Engineering
Mumbai, India

S. Nair
Assistant Professor
Dept. of Computer Engineering
D. J. Sanghvi College of Engineering
Mumbai, India

***Abstract*— The paper presents an empirical study of three text classification algorithms using two datasets. Naïve Bayes, Support Vector Machine and C4.5 have been compared by training the dataset instances on the Weka Tool. The two datasets are Diabetes and Calories. Diabetes dataset has a large number of training examples and attributes as compared to the Calories dataset. The results are compared based on the recall and precision values that each of the algorithms are returning. Another basis of comparison is the percentage split of the dataset into training set and test set. Results show that out of the three classifiers, SVM is computationally efficient. SVM has certain disadvantages which degrades its performance for small datasets. Thus, it is proposed that using Hybrid SVM may improve the existing drawbacks of SVM. Even if the approach with which SVM is applied on the dataset is changed, it can produce optimized results.**

Keywords—Support Vector Machine; Naive Bayes; Text Classification; Data Mining; C4.5; Weka

I. INTRODUCTION

Data mining is a procedure of discovering knowledge by analyzing data from different viewpoint and summarizing it into meaningful information. Text Classification is a supervised learning technique which is a sub-domain of data mining, used to assign text to classes from a predefined group of classes and has different applications such as spam filtering, sentiment analysis, language identification and genre classification. Our goal in this paper is to compare various text classification techniques according to different factors such as precision/recall, and percentage of correctly classified instances from the training set using the Weka Tool. Some of the classifiers that we have used in Weka are Support Vector Machine (SMO in Weka), C4.5 (J48 in Weka), and Naive Bayes. We used two datasets for comparison. First is the Calories dataset which consists of different food items and their characteristics as attributes and will classify the food items according to their distribution. The second dataset is the Diabetes dataset which would classify whether patients have diabetes or not. On training the datasets with different classifiers, it was observed that out of all the classifiers, SVM classified the instances with highest accuracy. Whilst SVM has its own disadvantages, such as kernel selection, it can be improved by combining SVM with various other algorithms.

The following is the flow of this paper: Section 2 presents an overview to the related work, while section 3 presents a walkthrough of text classification; section 4 concentrates on acquainting the user with the text classifiers we have used. Section 5 describes the datasets, section 6 presents the results and evaluation, and we conclude our work along with future works in section 7.

II. RELATED WORK

Many researchers have compared text categorization algorithms over different datasets. However, a comparison of classification algorithms over the calories dataset has not been attempted so far. Aggarwal [4] conducted a survey of text classification algorithms. Some of the classifiers surveyed by them were Probabilistic and Naive Bayes classifiers, rule - based classifiers, and Multinomial distribution. Likewise, Jimenez [5] demonstrated an example of text classification and clustering with Weka. A movie review dataset was classified into positive or negative reviews. The text of reviews was converted into vector format and Naive Bayes classifier was used for classification. Clustering was also carried out, and 65.25% instances were clustered correctly. Wilcox [6] applied classification algorithms to narrative reports. Methods such as Decision Trees, Bayesian classifiers were used to classify X-ray reports according to 6 attributes. They concluded that text classification algorithms were dependent on training set size. Similarly, Pandey [7] has reviewed text classification techniques for email filtering and management. Approaches used by them were Naive Bayes, Support Vector Machine (SVM), Decision Trees, Fuzzy Logic, etc. They concluded that for filtering, context based email-organization has the best potential.

III. TEXT CLASSIFICATION

Text classification [1] sorts documents into a fixed number of predefined categories. The documents can be multiple, unique or may not fit into a category at all. In that case, handling a large number of documents can become complicated. Thus, a text classifier places these documents into groups which are relevant to their content and makes it easier to sort them when a search for a specific document is carried out. The set of categories for the documents is called **Controlled Vocabulary [2].** A good analogy would be that of a student sorting a set of certificates, passport photocopies,

exam mark sheets and a few forms into different folders and labeling each folder according to its content for ease of retrieval later. A good text classifier though, would work efficiently for large training sets with several features. Feature Selection forms an integral part of any classification task and it is especially important in the case of text categorization because of the high dimensionality and presence of noise of features, so it is necessary to select only the most essential features. A common step of feature selection is stop-word removal and stemming. [8] Stop-word removal involves deleting words which are common and do not make much of a difference for classification. Stemming involves reducing words which are inflected to their "stem", the root word from which they derive. According to Basu [3], Text categorization requires, as a basis, the identification of features within the documents that can be used to discriminate amongst the documents and associate them to individual categories.

IV. Categorization Methods

A. Naive Bayes

According to Patra [8], Naive Bayes first learns training examples in priori probability when given unseen examples. The features are assumed to be independent meaning the presence of one feature does not affect the presence of another feature. Because of this assumption that attributes are independent of each other underlies on this approach, it is called 'Naïve'. Even though this theory violates the fact that attributes are dependent on each other, its performance is feasible. It is the most widely used classifier because of its simplicity and also because it is continuously adapting in case a user identifies an incorrectly classified example, thereby improving its efficiency. NB is based on the Bayes rule of conditional probability [16] given by formula (1). h is the hypothesis and x is the attribute.

$$P(h_i / x_i) = \frac{P(x_i / h) * P(h_i)}{P(x_i)} \quad (1)$$

B. C4.5

C4.5 is a modification of the ID3 algorithm which focuses on creating a decision tree, using a fixed set of attributes, to classify a training example into a fixed set of classes as stated by Macskassy et al [10]. C4.5 is an entropy based algorithm. It is a widely used decision tree learning algorithm. At every step, if the remaining instances all belong to the same class, it predicts that particular class, otherwise, it selects the attribute with the highest information gain and creates a decision based on that attribute to split the training set into one or two subsets. If the feature is discrete then the training set is split into one subset based on its discrete value. In the case of continuous features, two subsets will be created on the basis of threshold comparison. The above steps are repeated recursively till all the nodes are final, or until the threshold limit is met. The threshold limit will be specified by the user. Once the decision tree is built, C4.5 prunes the tree in order to avoid over fitting, again based on a setting specified by the user.

C. Support Vector Machine

SVMs are efficient binary classifiers that is based on structural risk minimization, meaning that it describes a general model of capacity control [11] and provides a trade-off between hypothesis space complexity (the VC dimension of approximating functions) and the quality of fitting the training data (empirical error). They are learning machines which are based on statistical learning theory. Any SVM would try to maximize the boundary between the positive and negative examples in a dataset. SVMs non-linearly map their n-dimensional input space into a higher-dimensional feature space. Using this high-dimensional feature space a linear classifier is then constructed with the help of quadratic programming, though this step can potentially be very costly. So to optimize this step, SVMs make use of different kernel methods which might improve the computation of inner numerical products.

V. The Datasets

A. Diabetes

This dataset consists of 768 instances with 9 attributes and the training examples are taken from a larger database which recorded the biological statistics of women, all around 21 years of age, and of Pima Indian origin. Given these training examples to a text classifier, the classifier will predict whether the patient has been tested positive/negative with diabetes mellitus based on the criteria set forth by the World Health Organization that a reading of 200 mg/dl, 2 hours post lunch shows signs of diabetes.

B. Calories

The dataset consists of 40 food items and 4 attributes. Some of them claim to be "lite", "low-fat", "no-fat", or "healthy" foods. These foods are classified based on their distribution i.e., nationally advertised, regionally distributed or locally prepared. Using the above three algorithms, the dataset is trained and correctly/incorrectly classified instances are determined by the Weka tool.

VI. Results and Evaluation

TABLE I. Diabetes Dataset Results

% Split of Training Set	Algorithm		
	Naive Bayes	*C4.5*	*SVM*
At 66% (261 instances)	77.01	76.24	79.31
At 90% (77 instances)	77.9	75.32	80.52
At 33% (515 instances)	73.98	70.29	75.73
Precision(Weighted Avg.)	0.767	0.756	0.787
Recall (Weighted Avg.)	0.77	0.762	0.793

Fig. 1. Correctly classified instance percentage after training for Diabetes.

TABLE II. CALORIES DATASET RESULTS

% Split of Training Set	Algorithm		
	Naive Bayes	C4.5	SVM
At 66% (14 instances)	78.57	78.57	71.52
At 90% (4 instances)	100	75	75
At 33% (27 instances)	81.48	85.18	66.67
Precision(Weighted Avg.)	0.844	0.802	0.81
Recall (Weighted Avg.)	0.786	0.786	0.714

Fig. 2. Correctly classified instance percentage after training for Calories.

In the first dataset, SVM outperforms the remaining two classifiers. Meanwhile, the performance of SVM is worse with the second dataset. Both the datasets were split into training set and testing set. When we select a 66% split, it implies that 66% of the dataset is training data, while the remaining instances are testing examples. It is observed that SVM performs poorly when the number of attributes is less which is evident in the Calories dataset.

VII. CONCLUSION AND FUTURE WORK

In a nutshell, text classification is an important area of research for applications requiring constant need to label documents and organize data for use in further research. Use of Naive Bayes, C4.5 and Support Vector Machine on a couple of datasets with varying training examples helped us compare performance of each of these classifiers. Support Vector Machine outperforms the remaining two classifiers and proves to be the best among the three. SVM may have some disadvantages but that can be improved by combining SVM with other algorithms. SVM has proven to be robust when the right parameters are chosen otherwise the results are not optimal. Sudheer et al [14] have proposed combining SVM with Particle Swarm Optimization for tuning the parameters. Another approach suggested by Phung et al [15] is to divide the Quadratic Programming problem into smaller sub-problems which will reduce computation time for large datasets.

ACKNOWLEDGMENT

We would like to thank our Honorable Principal Dr. Hari Vasudevan as well as our respected Head of Computer Department Dr. Narendra M. Shekokar at D. J. Sanghvi College of Engineering for their enlightening support. In addition, we would also like to thank SVKM for encouraging and enabling us to participate in such co-curricular events.

REFERENCES

[1] Joachims, Thorsten. Text Categorization with Support Vector Machines Learning with Many Relevant Features. Dortmund: Dekanat Informatik, Univ., 1997.

[2] Niharika, S., Latha, V. and Lavanya, D. (2012). A Survey on Text Categorization. International Journal of Computer Trends and Technology volume 3 Issue1 -2012.

[3] Basu, A., Walters, C. and Shepherd, M. Support vector machines for text categorization. p.7, 2003

[4] Aggarwal, C. and Zhai, C. A survey of text classification algorithms. Springer, pp.163—222, 2012.

[5] Jiménez, S. Text Classification and Clustering with WEKA, 2014.

[6] Wilcox, A. and Hripcsak, G. Classification algorithms applied to narrative reports. p.455, 1999.

[7] Pandey, U. and Chakraverty, S. A Review of Text Classification Approaches for E-mail Management. IACSIT International Journal of Engineering and Technology, 3(2), 2011.

[8] Patra, A. and Singh, D.(2013). A Survey Report on Text Classification with Different Term Weighing Methods and Comparison between Classification Algorithms. International Journal of Computer Applications Volume 75–No.7, August 2013 pp.14-18

[9] Korde, V. and Mahender, C. Text classification and classifiers: A survey. International Journal of Artificial Intelligence & Applications (IJAIA), 3(2), pp.85—99, 2012.

[10] Macskassy, S., Hirsh, H., Banerjee, A. and Dayanik, A. Converting numerical classification into text classification. Artificial Intelligence, 143(1), pp.51—77, 2003.

[11] Sewell, M. (2014). Structural Risk Minimization. [online] Svms.org. Available at: http://www.svms.org/srm/ [Accessed 4 Sep. 2014]

[12] Sebastiani, F. (2002). Machine learning in automated text categorization. ACM computing surveys (CSUR), 34(1), pp.1--47.

[13] Wahbeh, A. and Al-Kabi, M. (2012). Comparative Assessment of the Performance of Three WEKA Text Classifiers Applied to Arabic Text. Deanship of Research and Graduate Studies, Yarmouk University, Irbid, Jordan.

[14] Sudheer, C., Maheswaran, R., Panigrahi, B. and Mathur, S. (2014). A hybrid SVM-PSO model for forecasting monthly streamflow. *Neural Computing and Applications*, 24(6), pp.1381--1389.

[15] Phung, S., Nguyen, G. and Bouzerdoum, A. (2010). Efficient SVM training with reduced weighted samples.

[16] Dunham, M. (2003). *Data mining introductory and advanced topics*. 1st ed. Upper Saddle River, N.J.: Prentice Hall/Pearson Education.