Proceedings of the International Conference on Intelligent Sustainable Systems (ICISS 2017)
IEEE Xplore Compliant - Part Number:CFP17M19-ART, ISBN:978-1-5386-1959-9

A Machine Learning Approach to classify News Articles based on Location

Vignesh Rao
Department of Computer Science
Visvesvaraya National Institute of Technology
Nagpur,India
Email: raovignesh0210@gmail.com

Jayant Sachdev
Department of Computer Science & Engineering
Delhi Technological University
New Delhi,India
Email: jayantsachdev01@gmail.com

***Abstract*—With the influx of myriad news accompanied with busy lifestyle, there is a pressing need to classify news according to the requirements of an individual. People are generally more interested what is going on, in their immediate surroundings. In this paper, we model this problem by classifying the news articles based on cities and providing the entity with the collection of city specific news. We have developed our own web crawler for content extraction from the HTML pages of news articles. Random Forests, Naive Bayes and SVM classifiers have been employed and their accuracy has been noted. Results exhibit that machine learning techniques can be harnessed to achieve our goal and thus calls for further research to improve the efficiency of solving this issue.**

***Keywords*—Machine Learning, Random Forest Classifier, SVM, Naive Bayes, News classification,Natural Language Processing**

I. Introduction

There is a vast amount of information that is growing at an exponential rate everyday in the form of news articles. Digitization has led to an increasing number of people switching to online sources for daily news feeds. Commensurate to the advances in the digital era, people generally are preoccupied with hectic work-life and prefer to read articles pertaining to their interests [1]. Thus, most of the news articles, though informative, might be of less relevance to an individual. Hence, It poses a mammoth task for extracting relevant news with respect to an individual. The interests can depend on several factors like type of the news articles, place to which the news belongs to, etc. In this case, we have considered the interest based on geographical domain. For example, a person wants to read news specific to Mumbai and is provided with a flood of news relating to all the cities of India. In this case, it would be cumbersome for the person to find the city specific news.

In this research, we have implemented machine learning techniques to classify news articles belonging to a particular location. The location can be a city, state, country, etc but we have examined the results based on cities. The news articles from various websites like Indian Express, Hindustan Times,Times of India etc are extracted to form our dataset.

The underlying structure of the Web Page is the HTML language[2]. This contains boilerplate elements like navigation bars, advertisements, comment section, etc. Text classification method applied on this data directly would lead to a very less accuracy. The crux of solving this issue is designing a method to scrap out the clean news articles for further processing. For this we have designed a web crawler to crawl the set of news website and extract the main text out of the webpage. Further processing of the article involves tokenizing the text and deriving the stem of the individual tokens. This is followed by the removal of stop words. Finally, classification is performed and the trained classifier is used to predict the output class, in this case-city, of the input test article. Random Forest Classifier, Support Vector Machine and Multinomial Naive Bayes have been used in the classification phase.

The organization of the remaining part of the research paper is as follows. Section 2 is a brief description of classification methods used. The main methodology and our approach fills in the section 3. Experimental results are discussed and analyzed in Section 4. A succint summary is provided in Section 5.

II. Background

A. Random Forest CLassifier

Random Forest is a supervised classification algorithm which was first introduced properly in [3] by Leo Brieman. It uses a number of uncorrelated decision tree classifiers and fits them on various sub samples of dataset. Each sub sample dataset for decision tree has same size as the original dataset . Random forest Classifiers uses bootstrap aggregating techniques which repeatedly selects random sample of the training set with replacement and uses this sample to make trees learn since bootstrap technique decreases the variance and thus leads to better performance. Random Forest classifier is used in outlier detection and replacing missing data. It is scalable as it can run on large data sets.

B. Naive Bayes

Naive Bayes classifier is a popular method for text classification problems where given a document or article the classifier has to decide the category of the article. It was first proposed by D.Lewis [4]. Naive Bayes classifier is based on probabilistic technique of classification which derives its roots from Bayes Theorem. It is based on the assumption of independence between the various features. It is a scalable classifier and can run efficiently with large data sets. Naive Bayes classifier is fast as compared to other classifiers and thus is used as a baseline for text classification problems.

C. SVM Classifier

SVM(Support Vector Machine) [5] works on the principle of Supervised Learning. SVM requires a training set and labels associated with it. After training, if a test data is fed in, the model assigns it to one category or the other. It performs well with linear classification. It can even work efficiently on a non-linear classification using a kernel trick by mapping the inputs into high dimensional feature space. It constructs a hyper-plane for classification [6]. The hyper-plane is chosen such that the distance between the nearest data point on either side is maximized [12].

III. Methodology

The goal of our approach is to assign an output class to the news articles based on the content. Fig. 1 shows the entire flow of our process. The process starts with the Data Retrieval module which is the collection of our dataset wherein our self developed web scraping algorithm is employed to extract the actual text from the webpage. The dataset so formed is then divided into test data and train data. Train data forms 80% whereas test data forms 20% of the dataset collected. Data Preprocessing methods are applied to the train data and is used as an input to the classifier for training. The same data preprocessing methods are used for the test data and this acts as an input to the trained classifier which predicts the output class of the test news articles. This is followed by the evaluation of the accuracy of the trained classifier based on some performance metrices. Following sub sections provide further analysis of each of these components

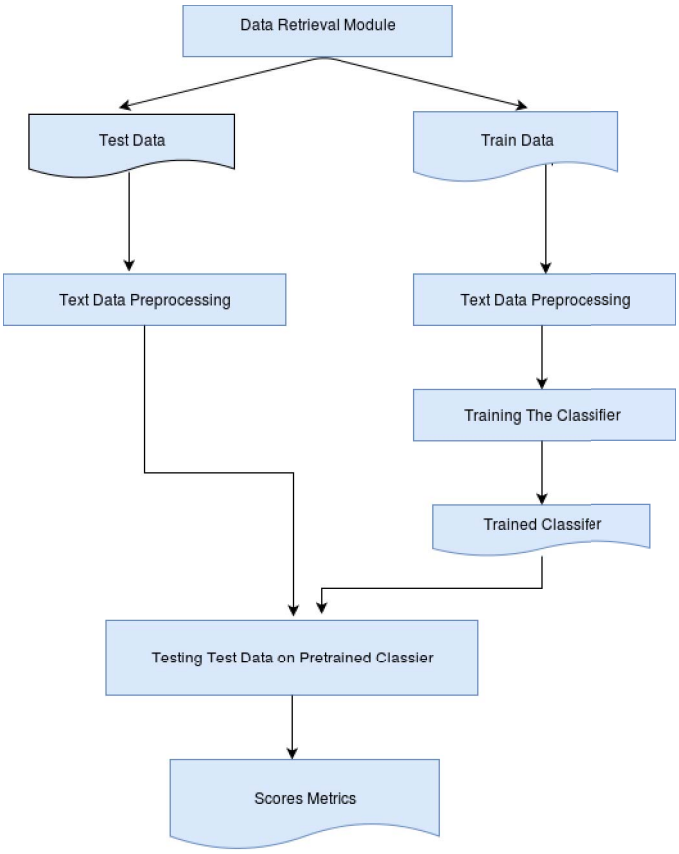

Fig. 1. Flow Chart Of the Process.

A. Data Retrieval

In this phase, news article data is being retrieved from various news websites. Fig.2 shows the process of data retrieval. The first step involves Parsing of RSS feeds from news websites. During RSS feed Parsing, links of various articles are being extracted. The second step of data retrieval process is collecting the retrieved URLS in a file which is to be used for further processing. The third step involves fetching of article text data from the collected URLS. Each URL is visited and article text is extracted from the HTML page of each news article. During this process RSS feeds for the 5 cities Delhi, Chandigarh, Kolkata, Mumbai and Lucknow were crawled from three news website Indian Express, Hindustan Times and Times of India . A total of 2000 articles were collected with a distribution such that the data set has 400 articles for each city. The equal distribution of dataset is ensured so that there is no bias in the training phase.

B. Text Pre-processing

This phase as shown in Fig.3 involves pre-processing of the fetched data. The first step involves tokenizing the articles in which the sequence of characters is converted into sequence of string which have identified meaning. Once the articles are tokenized the next step is stemming where every word in the article is reduced from its inflectional or derivationally related form to its base form[13]. It is followed by removal of stop words as these are the most common words and are of little significance in this classification process. For the removal of stop words, a list of specific stop words was created to be removed apart from the stop words provided by the nltk python package. The final step of this process is the Part of Speech Tagging where each word in the article is assigned a part of speech [11] such as noun, verb, adjective, noun plural etc.

In this process of classification Noun-singular (NN), Noun-plural (NNS), Proper nounPlural(NNPS) and Proper Noun Singular(NNP) tags are being kept and rest part of speech tags are being dropped.

C. Training a classifier

Training a classifier first involves the pre-processing module which extracts relevant part of the article as shown in Fig.4. This step is essential to improve the accuracy of the classifier. Input to the classifier is the training set and the set of labels corresponding to it. The processed news articles are labelled numerically based on the city tag which is pre-decided. Since input to the classifier are two vectors, the set of news articles and the labels need to be vectorized. After vectorization of these two entities, they act as an input to the classifier. Only 80% of the dataset is used for training the classifier, the rest is utilized for testing. Since training is performed only once, the classifier object which is trained is stored in a pickle file. Pickling is done to serialize the object and storing it into the disk for the testing phase. Similar process is applied for dumping the Count Vectorizer object for further use.

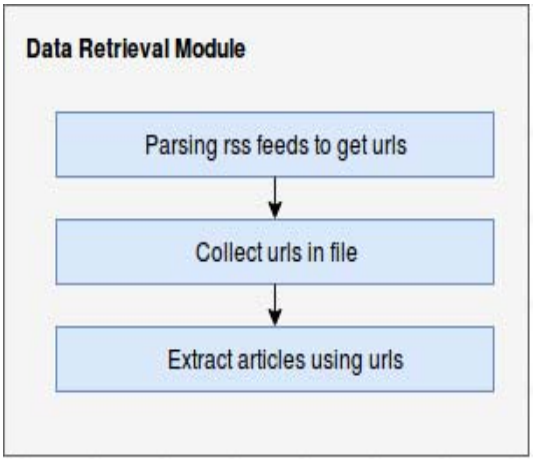

Fig. 2. Data Retrieval.

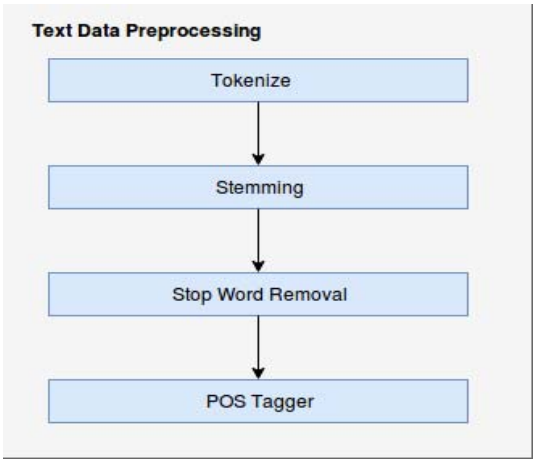

Fig. 3. Text Data Preprocessing

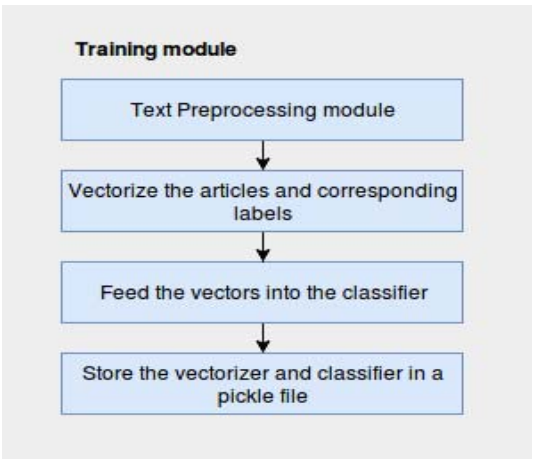

Fig. 4. Training

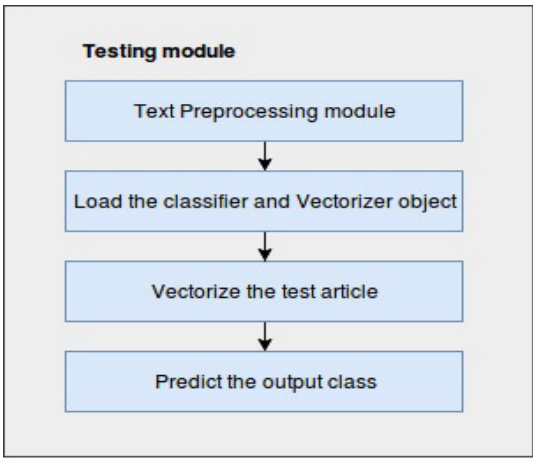

Fig. 5. Testing

D. Testing a classifier

The pickle file containing the stored classifier and the Count Vectorizer object is loaded. Since the stored classifier is trained, testing data is fed into it. The classifier predicts the class, in this case-city, of the corresponding article. The classification as performed by the trained classifier, acts a basis for determining the accuracy of the model. After testing, the accuracy of the classifier is noted based on various performance metrics like Precision, Recall and F1 score.

IV. RESULTS

In this experiment, we have used precision,recall and F1 score as the performance metrics as noted in table 6. F1 score is considered as the primary performance measure.

1. Precision:

Precision [10] is the ratio of the number of articles which are judged correctly(True Positive) to the total number of articles which the classifier predicted to belong to a particular category(True Positive and False Positive). It is also called as positive predictive value.

The precision can be defined as follows:

$$P = \frac{m}{m+n} \tag{1}$$

where, m stands for True Positive and n is the False Positive.

Classifier	Precision	Recall	F1-Score
Naïve Bayes	0.8341	0.825	0.8295
SVM	0.8026	0.775	0.7885
Random Forest	0.859276	0.845	0.8520

Fig. 6. Results table

2. Recall:

Recall [10]is the ratio of the number of articles which are judged correctly(True Positive) to the total number of articles which are actually in a particular category(False Negative and True Positive).

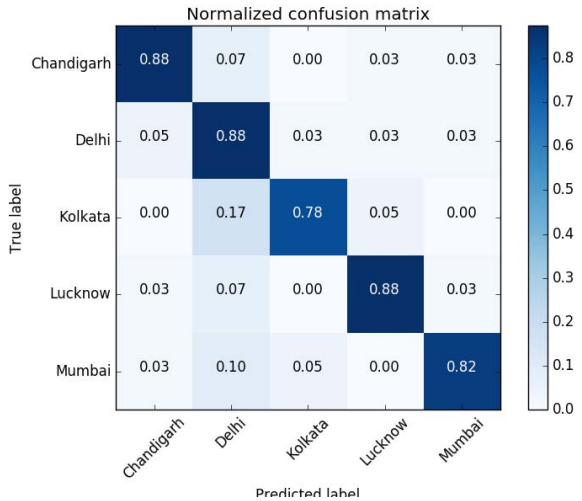

Fig. 7. Confusion Matrix For Random Forest.

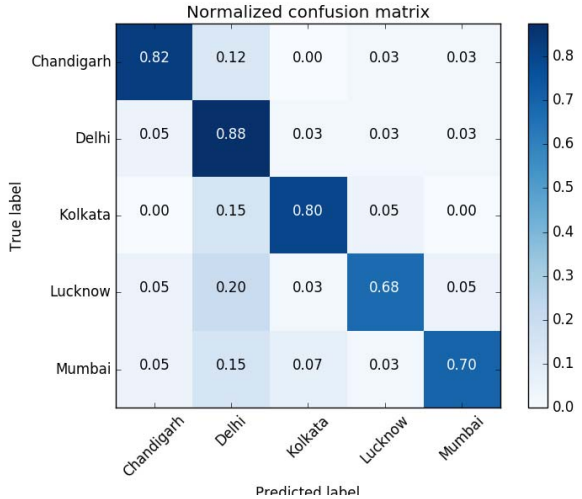

Fig. 8. Confusion Matrix For SVM.

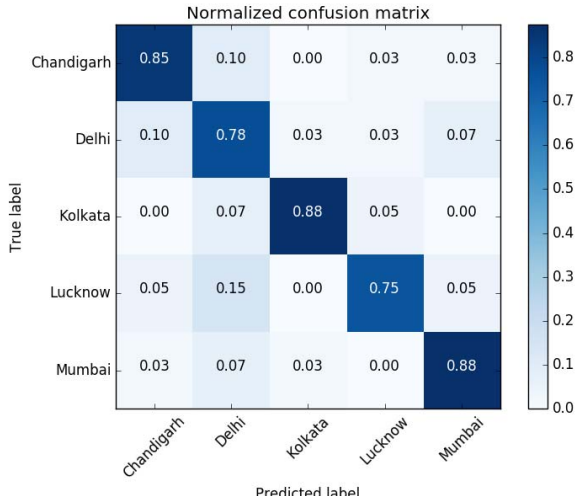

Fig. 9. Confusion Matrix For Naive Bayes.

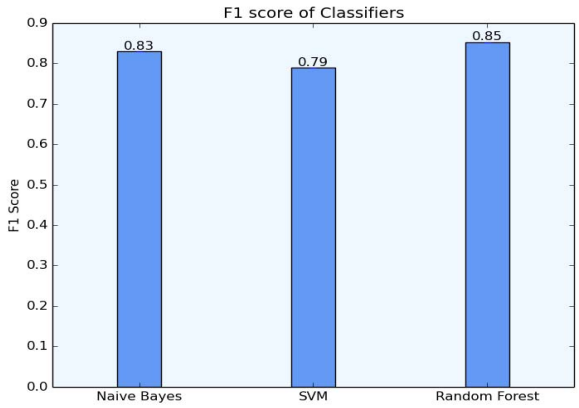

Fig. 10. Bar Plot.

The recall is defined as follows:

$$R = \frac{m}{m+t} \tag{2}$$

where, m stands for True Positive and t stands for False Negative.

It can be observed that Random Forest has performed the best with the highest F1 score of 85.20. Naive Bayes has also performed well with a F1 score of 82.95. Whereas, SVM is at the bottom with the F1 score of 78.85. The same order can be observed in case of precision and recall. The highest precision is observed in Random forest(85.92) and lowest in case of SVM(80.26). The highest and lowest values of Recall are 84.5 and 77.5 in Random Forest and SVM respectively. The confusion matrix has been plotted to visualize the true positives,true negatives,false positives and false negatives of our results.

3. F1 score:

F1 score considers both Precision and Recall of the test to calculate the score. It is the harmonic mean of the precision and recall [7]and is used majorly in [10] and [9]. The optimal value of F1 score is 1 and the worst is 0. Since F1 score combines precision and Recall into one value and it is effective to use one value as a measure [8] instead of both Precision and Recall, F1 score is used in this paper as the primary metric for comparing the effectiveness of the classifier.

$$F1 = 2 * (\frac{P * R}{P + R}) \tag{3}$$

where, P is the Precision and R stands for Recall.

V. Conclusion

In this paper, we have investigated the possibility to use machine learning algorithms to classify the news articles based on cities. The experiments show that this problem can be successfully solved by using various Classifiers such as Naive Bayes, Support vector Machines and Random Forest. Random Forest has outperformed the other classifiers. Naive Bayes has performed well too and Support Vector machine is at the bottom in terms of the performance metrics used in our

approach. The proposed system can be used as a part of more complex news article classification systems.

Our future target is to improve the accuracy and also try classifier like Neural Network. We can further increase the number of the input articles to 1000 fold compared to our present dataset for training our model inorder to improve on our results.

REFERENCES

[1] B. Pendharkar, P. Ambekar,P. Godbole, S. Joshi, and S. Abhyankar, *"Topic categorization of rss news feeds*," *G*roup vol. 4, p. 1, 2007.

[2] D. Shen, Z. Chen, Q. Yang, H. Zeng, B. Zhang, Y. Lu, and W. Ma,*Web-page classification through summarization*, in Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2004, pp. 242249

[3] Leo Breiman, *Random forests*, Machine Learning. vol. 45, no. 1, pp. 532, 2001.

[4] D. Lewis, *"Naive (bayes) at forty: The independence assumption in information retrieval*," *M*achine Learning: ECML-98, pp. 415, 1998

[5] J,.K. M. Han, *Data Mining:Concepts and Techniques*, 2nd ed. 2006.

[6] H. a. K. S. Yu, *"SVM tutorial: Classification, regression, and ranking*," *H*andbook of Natural Computing 2009.

[7] C. Rijsbergen, *Information Retrieval*,2nd ed. London: Butterworths, 1979.

[8] Y. Baeza and B. R. Neto, *Modern Information Retrieval.* Boston, 1999

[9] J. Davis and M. Goadrich, *The relationship between precision-recall and ROC curves*, in Proceedings of the 23rd International Conference on Machine Learning, ser. ICML 06. New York, NY, USA: ACM, 2006, pp. 233240 [Online]. Available: http://doi.acm.org/10.1145/1143844.1143874

[10] T. Landgrebe, P. Paclk, R. Duin, and A. Bradley, *Precision-recall operating characteristic (P-ROC) curves in imprecise environments*, in Proceedings of ICPR, 2006.

[11] Manning, C. and Schutze, H., *Foundation of statistical natural language processing* Cambridge,Mass:MIT press,1999.

[12] C. Ee and P. Lim, *Automated online news classification with personalization.*

[13] M.Kasthuri, Dr.S.Britto Ramesh Kumar, *A Framework for Language Independent Stemmer Using Dynamic Programming*, International Journal of Applied Engineering Research, ISSN 0973- 4562 Vol.10, pp 39000-39004, Number.18, 2015.