Text Document Classification Using Machine Learning and Optimization Techniques

Dr.S.V.Kogilavani, Dr.S.Malliga, Dr.C.S.Kanimozhiselvi
Assistant Professor(SRG), Professor, Associate Professor
Department of CSE, Kongu Engineering College, Perundurai, Erode, Tamil Nadu
Kogilavani.sv@gmail.com, mallisenthil@kongu.ac.in, kanimozhi@kongu.ac.in

Abstract: Text classification is the process of classifying the given text based on their categories. The goal of text classification systems is to increase discoverability of information and make all the knowledge discovered available or actionable to support strategic decision making. In this paper, a text classification system has been proposed for classifying the news articles based on their categories. In this system, the text data extracted from PDF document is converted into numerical feature vectors which will be further used for classification. Different machine learning classifiers like Naive Bayes and Support Vector Machine are used for classification. Performance of each classifier is evaluated based on evaluation techniques. Finally classifier performance is optimized using GridSearch. The result shows that the classifier performance is further improved using GridSearch.

Keywords: GridSearch , Machine Learning, Naïve Bayes, Support Vector Machine

I. INTRODUCTION

Data mining is the process of discovering patterns in large data sets involving methods at the intersection of machine learning, statistics, and database systems [1]. It is an essential process where intelligent methods are applied to extract data patterns [2]. The overall goal of the data mining process is to extract information from a data set and transform it into an understandable structure for further use. The actual data mining task is the semi-automatic or automatic analysis of large quantities of data to extract previously unknown, interesting patterns such as groups of data records (cluster analysis), unusual records (anomaly detection), and dependencies (association rule mining, sequential pattern mining).

Machine learning is a field of computer science that gives computer systems the ability to "learn" (i.e. progressively improve performance on a specific task) with data, without being explicitly programmed [3]. It explores the study and construction of algorithms that can learn from and make predictions on data. It is used to optimize performance criterion using example data or past experience.

Data mining task usually involves using machine learning techniques such Naive Bayes Classifier, Apriori algorithm, etc to perform classification of text data. The available dataset is split into training and test dataset. Training dataset is used to train the classifier and test dataset is used to validate the accuracy and performance of the classifier. In this work, Machine Learning algorithms can be used extensively to perform text classification.

II. LITERATURE REVIEW

Kabita Thaoroijam[4] describes that the huge growth of information and World Wide Web is no longer feasible for the user to understand all the data coming in or classify it into categories is tedious.

Text classification is a task of automatically sorting a set of documents into categories from a predefined set and is also one of important process in text mining.

Sundus Hassan, et al. [5] mentioned that the activity of labeling of documents according to their content is known as text categorization. Many experiments have been carried out to enhance text categorization by adding background knowledge to the document using knowledge repositories like Word Net, Open Project Directory (OPD), Wikipedia and Wikitology. Karishma Borkar1, et al.[6] presented that text classification is the undertaking of naturally sorting an arrangement of archives into classifications from a predefined set.

Content Classification is an information mining procedure used to anticipate bunch enrollment for information occurrences inside a given dataset. Duy Duc An Bui, et al.[7] describes that they used open-source tool to extract raw texts from a PDF document and developed a text classification algorithm that follows a multi-pass sieve framework to automatically classify PDF text snippets into Title, Abstract, Bodytext, Semistructure, and Metadata categories. To validate the algorithm, they developed a gold standard of PDF reports.

III. PROPOSED SYSTEM

Our proposed system has five modules: (a) Extraction of raw data from pdf, (b) Text data representation, (c) Machine Learning classifiers, (d) Performance Evaluation and (e) Gridsearch based classification.

The raw text is extracted from PDF document using a open sourced tool called PDFBox tool. The raw data is then stored in a text file. Then the features are extracted from the text files. The text files are converted into numerical vectors and used to apply the classification algorithm. The number of words in the document is counted and each word is assigned an integer id. Each unique word in the text file is considered as a feature. The counting returns a document-term matrix.

A document-term matrix or term-document matrix is a mathematical matrix that describes the frequency of terms that occur in a collection of documents. In a document-term matrix, rows correspond to documents in the collection and columns correspond to terms. There are various schemes for determining the value that each entry in the matrix should take. One such scheme is tf-idf [8]. Since counting the number of words gives more weightage to longer documents than shorter documents, we can use Term Frequency. The weightage of more common words which occurs in all document can be reduced. This is called as TF-IDF i.e Term Frequency Inverse Document Frequency.

There are various machine learning algorithms used for text classification. In this work, Naive Bayes classifier and Support Vector Machine are used for classification work. Firstly the Naive Bayes classifier is applied to the training data. It classifies the words into their corresponding categories and the performance of the classifier is evaluated by using the test data. Then Support Vector Machine is used for classification and its performance is also evaluated. The parameters of the classifiers can be optimized for better performance. Hence a tool called ‘GridSearchCV’ is used. Here only a few parameters of the classifier are chosen for optimization. Once optimized, we can find a difference in the performance of the classifiers. The mean accuracy of each experiment is calculated.

A. Raw Data Extraction

Classification of text data in a PDF document is complex. Hence they must be extracted. This is done by using a open-sourced tool called PDFBox tool. The tool extracts the raw text data from a PDF document and the contents are stored in a text file which can be used for further processing.

B. Text Data Representation

Text files contain words extracted from news articles. For text classification purpose, we convert the series of words into numerical feature vectors. Each unique word in the document is considered a feature. Simply counting the number of words gives more weightage to longer documents than shorter documents. Hence TF-IDF is used to reduce the weightage.

C. Machine Learning Classifiers

The core goal of classification is to predict category of the text [9]. Naive Bayes and Support Vector Machine were initially used to classify the news data. The goal is to classify a given document according to its category.

- *Naïve Bayes Classifier*

The Naive Bayes algorithm is an intuitive method that uses the probabilities of each attribute belonging to each class to make a prediction. Naive bayes simplifies the calculation of probabilities by assuming that the probability of each attribute belonging to a given class value is independent of all other attributes. This is a strong assumption but results in a fast and effective method [10].

- *Support Vector Machine Classifier*

Support Vector Machine (SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges [11]. We applied the machine learning classifiers to the training data. Then we evaluated the performance of the classification using test data.

(d) Performance Evaluation of classifiers

The test data is used to evaluate the performance of the machine learning classifiers. The accuracy of the Naive Bayes Classifier obtained is 77.38% and that of the Support Vector Machine is 82.38%.

(e) GridSearch based classification

Hyper-parameters are parameters that are not directly learnt within estimators. Any parameter provided when constructing an estimator may be optimized in this manner [12]. Various parameters of the machine learning classifier can be optimized by using GridSearchCV. A few parameters are taken for performance tuning and then the performance of the machine learning classifiers are evaluated to notice the difference in performance. The performance of the Naive Bayes Classifier increased to 90.6% and that of Support Vector Machine to 89.79%.

IV. PERFORMANCE EVALUATION

For experimentation, the famous 20 newsgroup dataset containing 20 newsgroups is used. Each newsgroup contains text files. Each text file contains text extracted from news articles. This data set consists of 20000 messages taken from 20 newsgroups. Each newsgroup is stored in a subdirectory, with each article stored as a separate file. The articles are typical postings and thus have headers including subject lines, signature files, and quoted portions of other articles [13].

The accuracy of the Naive Bayes Classifier obtained is 77.38% and that of the Support Vector

Machine is 82.38%. After using GridSearchCV to optimize the performance of the machine learning classifiers, the obtained performance value in Naive Bayes Classifier increased to 90.6% and that of Support Vector Machine to 89.79%.

Table 1. Performance evaluation of machine learning classifiers

Classifier	Accuracy
Naïve Bayes	77.38%
Support Vector Machine	82.38%
Naïve Bayes using Grid Search	90.6%
Support Vector Machine using Grid Search	89.79%

V. CONCLUSION

Text classification is a time consuming process. Classification of text data is difficult. Hence, the text data is represented as numerical feature vectors and used for classification. The text data is classified using Naive Bayes classifier and Support Vector Machine. The performance of classifier is optimized using Grid Search. The performance is found to have increased. Naive Bayes classifier and Support Vector Machine can be applied to different representations of the document like Bag of Words (BOW), Term Freqency (TF) and Term Frequency Inverse Document Frequency (TF-IDF). The mean accuracy of each experimentation can be calculated. Other evaluation metrics like precision, recall and accuracy can be used to compare the performance of classifier for each representation.

REFERENCES

1] https://en.wikipedia.org/wiki/Data_mining.
2] Han, Kamber, Pei, Jaiwei, Micheline, Jian,“Data Mining: Concepts and Techniques” ISBN 978-0-12-381479-1, 2011.
3] Samuel, Arthur, "Some Studies in Machine Learning Using the Game of Checkers", IBM Journal of Research and Development. **3** (3). doi:10.1147/rd.33.0210, 1959.
4] Kabita Thaoroijam, “A Study on Document Classification using Machine Learning Techniques”, IJCSI International Jornal Of Computer Science Issues, Vol 11, Issue 2, No 1, 2014.
5] Hassan, Sundus & Rafi, Muhammad & Shaikh, M., “Comparing SVM and Naive Bayes classifiers for text categorization with Wikitology as knowledge enrichment”, 10.1109/INMIC.2011.6151495, 2012.
6] Karishma Borkar1, Prof. Nutan Dhande2, “Efficient Text Classification of 20 Newsgroup Dataset using Classification Algorithm” ,International Journal on Recent and Innovation Trends in Computing and Communication, ISSN: 2321-8169 Volume: 5 Issue: 6 pp.1236–1240, 2017.
7] Duy Duc An Bui , Guilherme Del Fiol , John F. Hurdle, Siddhartha Jonnalagadda, “Extractive text summarization system to aid data extraction from full text in systematic review development”, Journal of Biomedical Informatics, v.64 n.C, p.265-272, December 2016.
8] https://en.wikipedia.org/wiki/Document-term_matrix
9] Carlos Guestrin, Emily Fox, Amazon Professor of Machine Learning, University of Washington, https://www.coursera.org/learn/ml-classification
10] Jason Brownlee, **“**How To Implement Naive Bayes From Scratch in Python”, December 8, 2014.

11] https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code
12] http://scikitlearn.org/stable/modules/grid_ search.html
13] https://archive.ics.uci.edu/ml/machine-learning-databases/20newsgroups-mld/20newsgroups.data.html