VJER-Vishwakarma Journal of Engineering Research www.vjer.in
Volume 1 Issue 4, December 2017 ISSN: 2456-8465

Popularity Prediction using Data Mining

Sanket Kulthe
Department of
Computer Engineering,
Vishwakarma Institute of
Information Technology,
Pune-48
Email: sanketkulthe81@gmail.com

Prashant Luhar
Department of
Computer Engineering,
Vishwakarma Institute of
Information Technology,
Pune-48
Email: prashant.luhar@yahoo.com

Anand Sonawane
Department of
Computer Engineering,
Vishwakarma Institute of
Information Technology,
Pune-48
Email: sonawaneanand95@gmail.com

Prof. Mrs. Leena A. Deshpande
Department of
Computer Engineering,
Vishwakarma Institute of
Information Technology,
Pune-48

Abstract—In recent years, there is tremendous growth in volume of text data available over the internet, digital libraries, news sources etc. with different context. Data Mining has been used in many types of software and devices in computer science field. These software and devices can make decisions with help of different data mining algorithms just like a human brain. Decision making mostly depends upon the training data given to the algorithm. Data plays a major and important part as one of the elements of Data Mining algorithm. We are basically going to classify the text data using classification algorithm.

Keywords—Data Mining, Classification, naive Bayes, Text Min-ing, Decision Tree, Feature Extraction.

II. Motivation

A lot of data is generated everyday, may it be digital data, text data or statistical data. Talking about text data, it is the perspective of people on different topics. This data is collected from various sources like social media sites, blogs, news media sites. We can classify this data and after analysis many predictions can be made. In case of governmental policies, people give their views and write it. After proper classification and analysis of this data, we can predict the popularity of policies and we can see that which policy had impact on the society.

III. Previous Work

.

S. M. Kamruzzaman et al (2005), they presented a new algorithm for text classification. Their algorithms uses less amount of documents for the training data. They proposed that instead of using word, association rules can be used to construct feature set. Then they used Naive Bayes on the feature set.

IV. Background Study

A. Data Mining

Data mining can be expressed as an abstraction of knowledge from the large data set. This knowledge can be used for the various fields. Data mining is fact finding for information. Number of databases are studied in data mining. There are following steps in KDD:(a) Data cleaning, (b) Data selection, (c) Data integration, (d) Data transformation, (e) Data mining, (f) Pattern evaluation, (g) Knowledge presentation. Data mining is the area in which predictive models can be built by using different computations. Data mining algorithms are used to build models which can further extract knowledge from data.

B. Text Mining

Text mining is subset of data mining which is used to gather knowledge from different pages containing textual data. Information Retrieval, abstraction, clustering, classification are fields in text mining. These text data contain too much information some of which is irrelevant and this can reduce the accuracy of the model. Some of the attributes may not contribute meaningful information the model. Some may devalue the quality and accuracy of the model. Irrelevant attributes simply add noise to the data and affect model accuracy. Noise increases the size of the model and the time and system resources needed for model building.

Data sets with more attributes may contain group of attributes that are correlated with each other. Wide data (many attributes) generally presents processing challenges for data mining algorithms. To minimize the effects of noise, cor-relation, and high dimensionality, some form of dimension reduction is a desirable preprocessing step for data mining. Feature selection and extraction are two approaches to dimen-sion reduction.

Feature Selection - Selecting the most relevant features. Feature Extraction - Combining attributes into a record set of features.

C. Feature Selection and Feature Extraction

Feature selection is a technique of selecting important attributes from data set. It is mainly used for three reasons:

Simplification of models to make them easier to interpret,

Shorter training

times, Generalization

Feature extraction is one of the dimensionality or attribute reduction process. Feature extraction creates new features or transforms the attributes from functions of the original features, whereas feature selection returns a subset of the features. The transformed features or attributes are combination of the origi-nal attributes. The feature extraction process results in a much smaller and richer set of attributes. Models built on extracted features may be of higher quality, because the data is described by fewer, more meaningful attributes. Feature extraction can also be used to enhance the speed and effectiveness of the model.

D. Classification

Classification is assigning a object to one or more classes. This can be done manually or automatically using algorithms. Classification is done mainly based on attributes, behavior or subject. The Classification problem can be illustrated as a training data set consisting of multiple records. Each record can bee searched by an unique record id, and includes of fields corresponding to the attributes. An attribute with a continuous domain is called a continuous attribute. An attribute with a finite domain of discrete values is called a categorical attribute. Categorical attribute is the classifying attribute and the its domain values are called class labels. Classification is the process of discovering a model for the class in terms of the remaining attributes. The objective is build a classifier model using the training data set, using this model we can classify the data which is not present in training data set. There are two forms of classification techniques currents in use:

1. Parallel Decision Tree Based:

It is better than the serial classification because it overcomes all the challenges faced in serial classification like handling larger datasets efficiently. Parallel algorithms are scalable in both runtime as well as memory requirements. Also parallel algorithms provide good speed up against serial ones.

2. Sequential Decision Tree Based:

This decision tree consists leaves and internal nodes. Each leaves present has class label assigned to it and every internal node has a particular decision assigned to itself. This type of classification is done in two steps:

i)Tree Induction - A tree is induced from the given training set.

ii)Tree Pruning - The previous tree is made more robust and accurate by amputating any type of statistical dependencies on a particular training data set.

E. Naive Bayes Classifier

This type of classification is based on Bayes Theorem. The performance of this classifier is comparable to other types like neural networks and decision tree classifiers. When applied on larger datasets, Bayesian Classifiers performs with high accuracy and speed. Naive Bayes classifier assumes that the effect of an attribute value on a given class is independent of the values of the other attributes. This is also known as condi-tional independence. When applying

Naive Bayes classifier to classify text data, each word position in a document is defined as an attribute and the value of that attribute to be the word found in that position.

V. PROPOSED WORK

In this project, we are proposing weight based feature extraction to remove irrelevant features and increase the accuracy for classification algorithm. Different weights are assigned to different keywords accordingly to its importance in defining class of text. For example, when a given tweet has a word which matches with our given string of words certain weight is added to the class labels for that tweet. Accordingly, weight of all class labels for that raw text is calculated and the class label with maximum weight is considered for that raw text. This way of classification makes algorithm more accurate as matching is done using the features extracted from the training set of data. These extracted features are helpful for giving weights to features of a given raw text.

We have collected the data from various sources available like Twitter, Facebook, Blogs etc. The data collected was raw data, having many attributes, some of which were irrelevant and was in different forms. The main aim was to collect the data and store it in same format. After collecting we needed to preprocess the data. Preprocessing the data was very important task because raw data has many errors and outliers which can reduce the accuracy and efficiency of the classifier.

The working for Text Classification

The proposed algorithm has various modules for classifying text. These modules are described below:

A. Data Collection

The data is collected form twitter using POST API. The collection service is provided by twitter officially. The data is collected in form of JSON format. The collection requires authentication and verification keys. Along with this Twitter Archiver tool is used to collect data it is GUI tool which records data on google sheets.

B. Data Cleaner

To extract features from raw text, we first have to prepare it. In this step, unnecessary words and symbols are removed. Keywords like is, am, are, to, the, a, an etc. are dropped and also the punctuations and stop-words. This cleaned text is used to extract keywords which will be used for feature extraction and also classification.

We can take example of Twitter data related to #MakeInIndia: Raw Data: Most sophisticated #ATAG's designed by @DRDO India & #MakeInIndia set to meet the operational shortfall of defence forces shortly.

Preprocessed Data: sophisticated atags designed drdoindia makeinindia set meet operational shortfall defence forces shortly.

C. Data Normalization

The cleaned data is normalized and converted into uniform format. The text data is converted to small caps and whole data is converted to UTF-8 format.

D. Feature Extraction

The data set of a particular class contains a set of features. The features are represented by n-grams. The terms that represent the features strongly are used for classification of the unknown data. For example: the term CleanIndia can be used to identify the text belonging to SwachhBharat class.

E. Assigning weights to the feature terms

The feature terms are assigned some weight according to their importance in a class. The more important terms are given greater weight than the rest of the terms. We then created a reference dictionary of the terms and relative weights.

F. Classifying the unknown data

Whenever unknown data arrives the particular tweet text is compared to the reference dictionary and the weight is measured. The comparison with reference dictionary which gives maximum weight is used as class label for that tweet text.

The results of our model can be collected and represented using a graph or similar format. Different graph plot libraries can be used for the same.

Tweet Text [Cleaned]	DEM_Weight	Swachh_Weight	MKI_Weight	Class Label
10000 blackmoney httpstcovhc0plnrzn	11	0	0	DEM
makeinindia indias chemicals sector winning	0	0	24	MKI
makeinindia india records highest ever yearo	0	0	22	MKI
small idea swachhbharat cleanindia pmoindia	0	12	0	SWACHH
theofficialsbi swachhbharat swachhbharat	0	8	0	SWACHH
swachhbharatgov thinkeatsave lets try reduc	0	16	0	SWACHH

This is an example where we can see classification of according to class labels. Here, "DEM" indicates Demonetization class label, "Swachh" indicates Swachh Bharat class label and "MKI" is Make In India class. Weight of any tweet is calculated for every class label. Whichever weight is maximum from thee three, the tweet belongs that class label, if the maximum value crossess some theshold value which can be altered accordingly. This method properly classifies the given text data in their particular class labels. Example: As we can see in table, first tweet is classified in Demonetization class. We can confirm same by analyzing the tweet text.

VI. CONCLUSION

In our project we require accurate classification of data for further analysis. For classification initially we implemented Naive Bayes classification and support vector machine algorithm for classification on test data set. However the results of these classification approaches were inaccurate and this could have negative effect on analysis part. Because of this we

implemented our method to classify the data which we found to be more accurate in our application.

VII. ACKNOWLEDGMENT

We would like to extend our acknowledgement to all the people who have been very helpful and without whom this project would not have been completed.
We express our profound gratitude to reviewers of this paper. We would also like to thank NCPCI team for allowing us to present this paper.
We also wish to express our deep sense of gratitude to Prof. Dr. Sachin Sakhare, HOD of Computer Department for able guidance and support.
Last but not the least, We would like to thank all our class and friends, all the teaching and non-teaching staff members for their valuable support in the development of the project.

REFERENCES

[1] Dr. S. Vijayarani et al, Preprocessing Techniques for Text Mining - An Overview, Bharathiar University, Coimbatore, Tamilnadu, India, 2016.
[2] Anuradha Purohit et al, Text Classification in Data Mining, International Journal of Scientific and Research Publications, Volume 5, Issue 6, June 2015.
[3] William B. Cavnar et al, N-Gram-Based Text Categorization, Environmental Research Institute of Michigan, Ann Arbor MI 48113-4001.
[4] Menaka S and Radha N, Text Classification using Keyword Extraction Technique, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 12, December 2013.
[5] Prof Sukhjit Singh Sehra, A review paper on Algoritms used for Text Classification, IJAIEM, Volume 2, Issue 3, March 2013.
[6] S. M. Kamruzzaman et al, Text Classification Using Data Mining, ICTM 2005.
[7] Shweta Kharya et al, Weighted Naive Bayes Classifier, International Journal of Computer Applications, Volume 133 No.9, January 2016.
[8] Arjun Srinivas Nayak et al, Survey on Pre-Processing Techniques for Text Mining, International Journal Of Engineering And Computer Science, Volume 5 Issues 6 June 2016.