Efficient Text Mining Using Side Information of Documents

[1]Rosemary Tripura, [2]P.Selvaraj
[1] Student, [2] Assistant Professor
[1]Dept. Of Information Technology, SRM University, Kattankulathur-603203, India
[2]Dept. Of Information Technology, SRM University, Kattankulathur-603203, India

***Abstract* -Due to the increasing availability of digital data, text document continue to grow as well hence the need of text mining. These digital documents comprise of the normal body text as well as side information. The side information will be in different formats for example hyperlinks and may contain useful information for mining. It is of utmost importance that the value of the side information be ascertained before consideration in the data selected for the text mining process as it may give an adverse impact on the quality of text mined. A principled way to perform the mining process is therefore required so as to maximize on the benefits of side information. In this paper, we use the Naive Bayes model to create an effective text mining approach.**

***Index Terms* -data mining, text mining, Stop word, word stemming, NLP.**

I. INTRODUCTION

Data mining and knowledge discovery have evolved in several disciplines such as database technology, machine learning, artificial intelligence, neural networking, etc. [1] Due to the rapid growth and ubiquity of digital data, a lot of attention and effort has been drawn to improving data mining techniques. Data mining is defined as the practice of automatically searching large stores of data to discover patterns and trends that go beyond simple analysis. [2] Knowledge discovery can be viewed as the process of nontrivial extraction of information from large databases, information that is implicitly presented in the data, previously unknown and potentially useful for users, with data mining as one of its essential steps.

Previous publications have proposed a number of data mining techniques that have been put to use to improve several knowledge discovery tasks. Text mining, also known as Intelligent Text Analysis, Text Data Mining or Knowledge-Discovery in Text (KDT) [4], is a variation in the field called data mining that tries discovery by computer of new, previously unknown information, by automatically extracting information from different written resources [3]. Sebastiani describes text mining as a system that analyzes large quantities of natural language text and detects lexical or linguistic usage patterns in an attempt to extract (probably) useful information from unstructured textual data through the identification and exploration of interesting patterns [12].Finding accurate knowledge or features in text documents so as to help users find what they want is quite a challenging process. Moreover the data sets considered in text mining can be either structured or unstructured such as emails, full-text documents and HTML files etc.

This paper focuses on improving text data mining through the use of side information. In text documents, side information is available in different forms such as; document origin information, user-access behavior from web logs, the links in the document, or other non-textual attributes, etc. all the forms of side information contain large volumes of clustering data. However, the relative importance of this side information could also be tough to estimate, especially when a number of the data is noisy. In such cases, it may be risky to include side-information into the mining method, because it will either improve the standard of the illustration for the mining method, or will add noise to the method, therefore, a principle is needed to perform the mining method,[10][13] that maximize the benefits from victimization aspect of data.

This paper proposes the usage Naïve Bayes classifier algorithms to improve text mining process from the document together with its associated side information.

.

II. BACKGROUND

***Natural Language Processing*:** It involves reducing the text; by reducing the original term-by-document matrix with a much smaller matrix. Unimportant words in the document get discarded at this stage. The text is analyzed against the natural language (human speech) structures. The system performs grammatical analysis on each sentence, singling out the most relevant or important words from both the side information and the original body.

***Information Extraction*:** This involves structuring the data generated by the Natural Language Processing. It is done by clustering, classification and predictive methods that are employed while carrying out the mining process.

III. RELATED WORK

Several studies have been carried in recent years on the problem of clustering in text collections [13], [19], [21], in the database and information retrieval communities. However, this work is primarily designed for the problem of pure text clustering, in the

absence of other kinds of attributes. In many application domains, huge amount of side-information is also associated along with the documents. Scatter-gather technique is one of the most well known techniques for text-clustering [24], which uses a combination of agglomerative and partition clustering. In [22], [23], other related methods for text-clustering which use similar methods are discussed.
The problem of text clustering has also been studied in context of scalability in [25], [24], [26]. However, all of these methods are designed for the case of pure text data, and do not work for cases in which the text-data is combined with other forms of data. Also [10], [11], [12], [5], [17] proposed approaches for the utilization aspect of document data for mining the Text knowledge. In survey of text classification algorithms, they use generalization and suppression techniques to safeguard the data [14], [12]. The data in the system is analyzed for generalization like replacing (or recoding) a value with a less specific but semantically consistent values. By using generalization and suppression techniques the data can be secured and semantically having consistent values. The major issue in this is that, there is no clear explanation on how the data is going to be secured in suppression technique. Considering, the data is not semantically linked. In that case, this technique won't be effective.
In order to achieve a more efficient text mining, this study will adopt an algorithm Naïve Bayes model to create an effective text mining approach.

A. Motivation

Sophisticated as they are, computers still have a long way to go in comprehending the natural language, which is a major problem in text mining. Humans have the ability to distinguish and apply linguistic patterns to text and can easily overcome obstacles that computers cannot handle such as slang, spelling variations and contextual meaning. However, although our language capabilities allow us to comprehend unstructured data, we lack the computer's ability to process text in large volumes or at high speeds. In the case of documents that contain side information, an approach which carefully ascertains the coherence of the clustering characteristics of the side information with that of the text content [5]. This is due to the fact that although useful in some situations, it can be quite noisy enough to adversely affect the quality of the overall mining process.

IV. PRELIMINARIES

A. Naïve Bayes algorithm:

The algorithm is a simple but important probabilistic model that is used for text classification. It combines prior knowledge with observed data on its likelihood given some training data. It then computes the maximum a posterior (MAP) hypothesis or the Maximum Likelihood (ML) hypothesis. [8][9]The algorithm assumes conditional independence between attributes and assigns the MAP class to new instances. This procedure is based on Bayes Rule, which says: if you have a hypothesis hand data D which bears on the hypothesis, then:

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

P(h): independent probability of h: prior probability
P(D): independent probability of D
P(D|h): conditional probability of D given
h: likelihood conditional probability of h given D: posterior probability
Computing the maximum a posteriori hypothesis for the data:

$$h_{MAP} = \underset{h \in H}{argmax}\ (h|D)$$

$$= \underset{h \in H}{argmax} \frac{P(D|h)P(h)}{P(D)}$$

$$= \underset{h \in H}{argmax} P(D|h)P(h)$$

Where; *H* is a set of all hypotheses
The Maximum Likelihood hypothesis:
Assuming that all hypotheses are equally probable a priori, This is called assuming a uniform prior, which simplifies the posterior computation:

$$h_{ML} = \underset{h \in H}{arg\max} P(D|h)$$

The classifier:
Assuming that the training set consists of instances described as conjunctions of attribute values, the target classification is based on finite set of classes V. The learner's task is to predict the correct class for a new instance $\langle a_1, a_2, \ldots, a_n \rangle$.
The idea is to assign most probable class using the Bayes Rule.

$$v_{MAP} = \arg\max_{v_j \in V} P(v_j|a_1, a_2, \ldots, a_n)$$

$$= \arg\max_{v_j \in V} \frac{P(a_1, a_2, \ldots, a_n|v_j)P(v_j)}{P(a_1, a_2, \ldots, a_n)}$$

$$= \arg\max_{v_j \in V} P(a_1, a_2, \ldots, a_n|v_j)P(v_j)$$

Parameter estimation:

The relative frequency of each target class in the training set has to be computed to estimate$P(v_j)$. Estimating $P(a_1, a_2, \ldots, a_n|v_j)$ is generally difficult because there are typically not enough instances for each attribute combination that is part of the training set thus the sparse data problem.

$$P(a_1, a_2, \ldots, a_n|v_j) = \prod_i P(a_i|v_j)$$

Hence we get the following classifier:

$$v_{NB} = \arg\max_{v_j \in V} P(v_j) \prod_i P(a_i|v_j)$$

V. PROPOSED APPROACH:

To achieve a more efficient text mining, the proposed system will adopt the Naïve Bayes algorithm for text classification.

A. System overview

It involves preprocessing text so as to distill the document into some structured format, reducing the results into a more practical size and finally mining the reduced data with a traditional data mining technique.

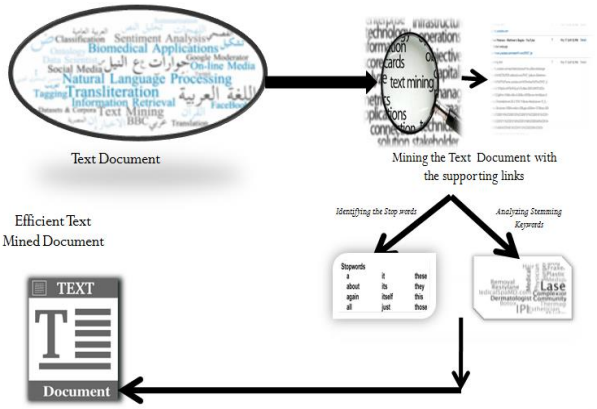

Figure 1 Overview of the System

As shown in Fig 1, the text document has to be chosen first, and then we have to extract its text and structure. Data parsing is done and from there, considering each and every sentence, stop words such as "the", "at", etc, have to be removed, while stemming expanded words as well.

B. Detailed description of the proposed system

The proposed system is implemented as follows:

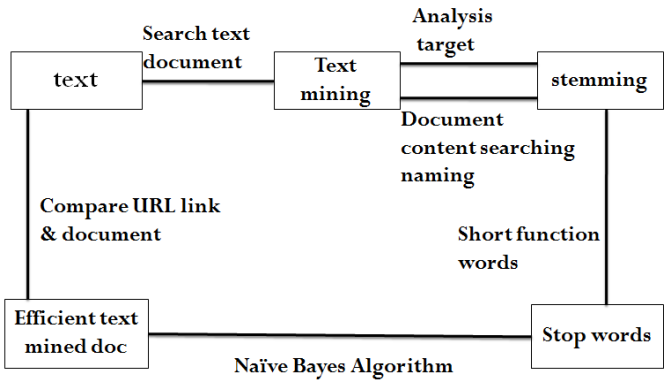

Fig 2 System Flowchart

Text pre-processing:

A document of any size is uploaded; it can be structured or unstructured. Text data is basically unstructured. The side information is extracted also at the same time. To make text data useful, unstructured text data is converted into structured data for further processing, this process of preprocessing consists of several steps:

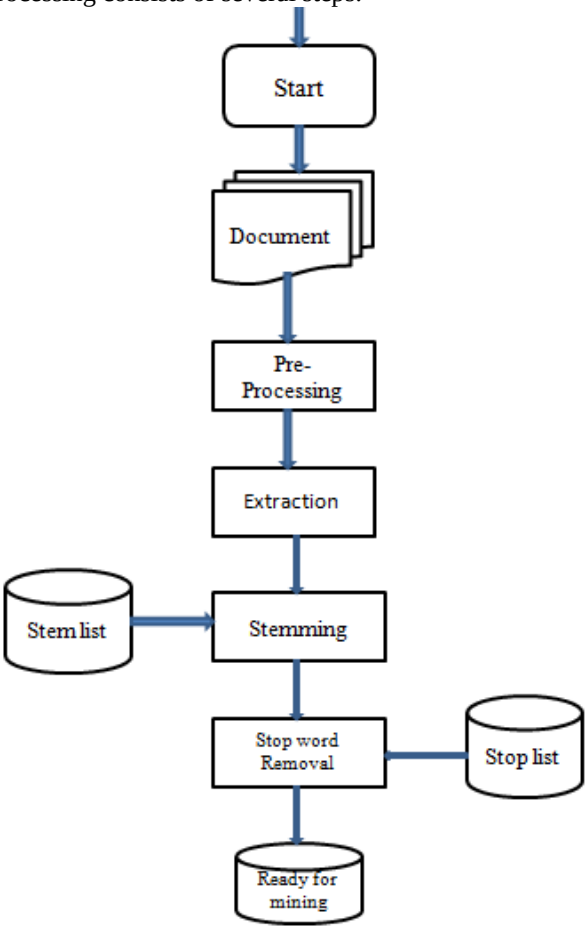

Figure 3: Text Processing

Data parsing:

Parsing text involves identifying the spaces, punctuation, and other non-alphanumeric characters found in text documents, and separating the words from these other characters. Most programming and statistical languages contain character procedures that can be used to parse the text data. It includes two processes; fetching the data to be mined and diving it or selecting sentences from it. Sentences are extracted from the document by weighing their importance based on the relevance to the topic of mining. After all the punctuation marks have been removed the process of data parsing can be best represented by the following algorithm:

(1) Initialize an array to hold the words that are parsed.
(2) Search for the first space and record its position.
(3) Extract the string from the first position to the position before the next space. This is the first word.
(4) Find the next space and record its position.
(5) Use the positions of the previous and subsequent space to extract the second word
(6) Repeat until all words are parsed.

To improve the speed of data parsing we use the split function, this splits words from spaces and other unwanted characters. A character string is taken as input then it is split in separate words.

Stemming:

The stemming method is used to find out the stem or root to a given word. When stemming a word, the "stem" or main root of the word replaces the numerous grammatical permutations of the word, such as plural versus singular, past and futures tenses, etc. As part of the process of stemming, synonyms are generally replaced with a single term. For instance, in the survey data, some respondents use the abbreviation ERM when referring to enterprise risk management. For example, the word "hand" can be stemmed from handle, handling, handful, etc. the purpose of stemming is to reduce the number of words by removing various suffixes, and thus presenting or storing only the stems to save memory and processing time.

Removing stop words:

This step involves eliminating articles and other words that convey little or no information. The most frequently used words in English are useless when it comes to Text Mining. Such words are called Stop words. Stop words are Language-specific be in form of pronouns, prepositions, conjunctions, etc. Then a table where each term in the text data becomes a variable with a numeric value for each record is created, we then text mine the processed data.

Efficient mined Document:

After the stemming and stop word analysis, the common words are analyzed from number of occurrences, and most common words get eliminated to make the efficient mined document. Data tokenize split is to split up the words and gets separating the verbs and noun to consolidating the meaningful required information from the given document and the fetched web page.

VI. RESULTS

Our system is implemented on two types of documents a Microsoft Word document with hyperlinks, for side information and an XML document.

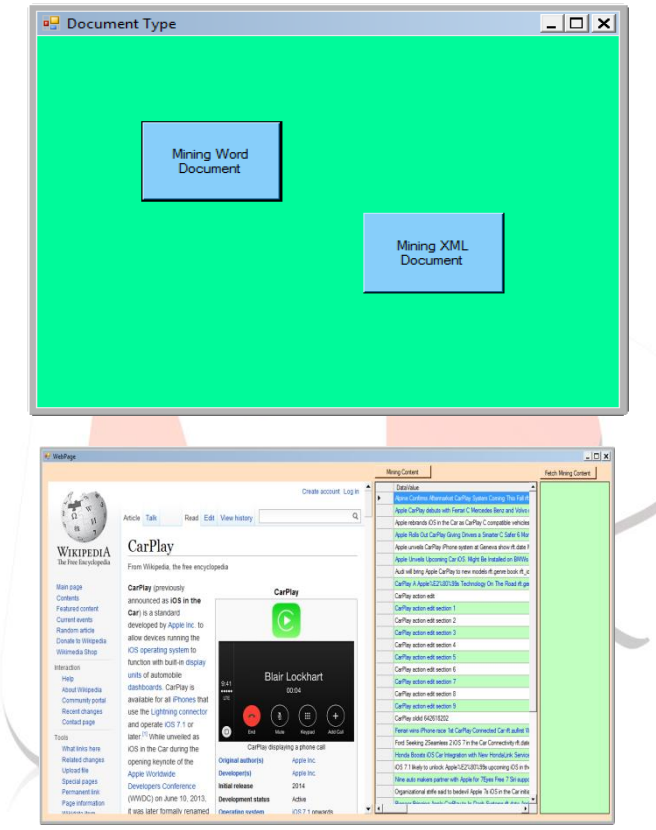

Fig 4 System Output

VII. CONCLUSION & FUTURE WORK

This paper acknowledges the fact that side information present in text documents be useful or may cause damage to the quality of mined text. Therefore the proposed technique of Naïve Bayes Text Mining classifier algorithm is use in order to provide more efficiency to the mined documents. The proposed technique allows side information to be considered also during the mining process. The proposed system uses the concept of website URL crawling to extract the information of the document and search priority is given to the words in the document and side information to crawl relevant websites. In future using the implemented level of the current system, to collect an efficient mined document by comparing multiple URL'S from the multiple web pages to consolidate the effective text mined content.

REFERENCES

[1] Feldman, R., Sanger, J., The Text Mining Handbook. Cambridge University Press, 2007

[2] NingZhong, Yuefeng Li, and Sheng-Tang Wu. Effective Pattern Discovery for Text Mining.

[3] Berry Michael W. "Automatic Discovery of Similar Words", in "Survey of Text Mining: Clustering, Classification and Retrieval", Springer Verlag, New York, LLC, 24-43 2004.

[4] Subamanikandan A, Arulmurugan R, On the Use of Side Information for Text Mining using Clustering and Classification Techniques-A Survey, Bannari Amman Institute of Technology, Anna University, Bannari Amman Institute of Technology, Anna University, Sathyamangalam, Erode, India.

[5] Mr. Y.R. Gurav, Assoc. Prof. P.B. Kumbharkar, A Review on Side Information Entangling For Effective Clustering Of Text Documents in Data Mining, Computer Engineering, CAYMET' Siddhant College of Engineering, Sudumbare, Pune, Maharashtra, India.

[6] Goldberg, D.E., Genetic algorithms in search, Optimization, and machine learning. Reading, MA: Addison Wesley, 1989.

[7] Zitzler, E. Evolutionary Algorithms for Multi objective Optimization: Methods and Applications. Ph.D. thesis, Swiss Federal Institute of Technology (ETH) Zurich, Switzerland. Shaker Verlag, Aachen, Germany, ISBN 3-8265-6831-1.1999

[8] Mitchell, Tom. M, Machine Learning. New York: McGraw-Hill, 1997.

[9] Ian H., and Eibe Frank, Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. San Diego, CA: Morgan Kaufmann, 2000.

[10] C. C. Aggarwal and H. Wang, Managing and Mining Graph Data. New York, NY, USA: Springer, 2010.

[11] C. C. Aggarwal and C.-X. Zhai, Mining Text Data. New York, NY, USA: Springer, 2012.

[12] C. C. Aggarwal and C.-X. Zhao, "survey of text classification algorithms," in Mining Text Data. New York, NY, USA: Springer, 2012.

[13] C. C. Agawam and P. S. Yu, "A framework for clustering massive text and categorical data streams," in Proc. SIAM Conf. Data Mining, 2006, pp. 477–481.

[14] C. C. Agawam, S. C. Gates, and P. S. Yu, "On using partial supervision for text categorization," IEEE Trans. Know. Data Eng., vol. 16, no. 2, pp. 245–255, Feb. 2004.

[15] C. C. Agawam and P. S. Yu, "On text clustering with side information," in Proc. IEEE ICDE Conf., Washington, DC, USA, 2012.

[16] R. Angel ova and S. Siersdorfer, "A neighborhood-based approach for clustering of linked document collections," in Proc. CIKM Conf., New York, NY, USA, 2006, pp. 778–779.

[17] A. Banerjee and S. Basu, "Topic models over text streams: A study of batch and online unsupervised learning," in Proc. SDM Conf., 2007, pp. 437–442.

[18] J. Chang and D. Blei, "Relational topic models for document networks," in Proc. AISTASIS, Clearwater, FL, USA, 2009, pp. 81–88.

[19] D. Cutting, D. Karger, J. Pedersen, and J. Tukey, "Scatter/Gather: "cluster-based approach to browsing large document collections," in Proc.ACM SIGIR Conf., New York, NY, USA, 1992, pp. 318–329.

[20] I. Dhillon, "Co-clustering documents and words using bipartite spectral graph partitioning," in Proc. ACM KDD Conf., New York, NY, USA, 2001, pp. 269–274.

[21] M. Steinbach, G. Karypis, and V. Kumar, "A comparison of document clustering techniques," in Proc. Text Mining Workshop KDD, 2000, pp. 109–110

[22] H. Schutze and C. Silverstein, "Projections for efficient document clustering," in Proc. ACM SIGIR Conf., New York, NY, USA, 1997,pp. 74–81.

[23] C. Silverstein and J. Pedersen, "Almost-constant time clustering of arbitrary corpus sets," in Proc. ACM SIGIR Conf., New York, NY, USA, 1997, pp. 60–66.

[24] Q. He, K. Chang, E.-P. Lim, and J. Zhang, "Bursty feature representation for clustering text streams," in Proc. SDM Conf., 2007, pp. 491–496.

[25] P. Domingo's and M. J. Pazzani, "On the optimality of the simple Bayesian classifier under zero-one loss," Mach. Learn., vol. 29, no. 2–3, pp. 103–130, 1997.

[26] S. Zhong, "Efficient streaming text clustering," Neural Network., vol. 18, no. 5–6, pp. 790–798, 2005.