A Novel Approach for Text Categorization of Unorganized data based with Information Extraction

Suneetha Manne
Assistant Professor , Department of IT
VRSEC, Vijayawada
Andhra Pradesh
suneethamanne@gmail.com

Dr. S. sameen Fatima
Professor and HOD, Department of Computer Science Engineering
Osmania University, Hyderabad
Andhra Pradesh
sammenf@gmail.com

***Abstract*—Internet has made a profound change in the lives of many enthusiastic innovators and researchers. The information available on the web has knocked the doors of Knowledge Discovery leading to a new Information era. Unfortunately, most Search Engines provide web content which is irrelevant to the information intended to the browser. Many Text Categorization techniques for web content have been developed, to recognize the given document's category but failed to make trust worthy results. This paper primarily focuses on web content categorization based on classic summarization technique by enabling the classification at word level. The web document is preprocessed first which involves filtering the content with classical techniques and then is converted into organized data. The organized data is then treated with predefined hierarchical categorical set to identify the exact category.**

Keywords-Text Categorization, Text Mining, Information Extraction, Feature Term Extraction, Information Retrieval, Pyramidal Model, Term Frequency.

I. INTRODUCTION

It is observed that people involved in research study need to analyze the research papers, e-books and other resources available on web. The same is the situation where a doctor finds difficulty in comparing the symptoms of a cancer patient to the already available categories, to recognize the stage he/she is suffering. Even in the Human Resources department of Multinational companies, it is difficult to categorize the Curriculum Vitae received from hundredths or often thousands of applicants. In such applications Text Categorization has become the key tool for automatic handling and organizing of text information.

Text categorization is the task of classifying a document under a predefined category. More formally, if *d*i is a document of the entire set of documents *D* and {*c*1, *c2*, *cn*} is the set of all the categories, then text categorization assigns one category *c*j to a document *d*i. For example, a document contains the terms like Decision rules, information retrieval, summarization, Bayes, nearest, cleaning, frequency, preprocessing, test, trained data etc., has more possibility to fall under the category called –"*Data Mining*". But when we are dealing with hyper text data, we need to include lot of preprocessing techniques so as to make the classification efficient. In Figure 1 is given the graphical representation of the text categorization process [15].

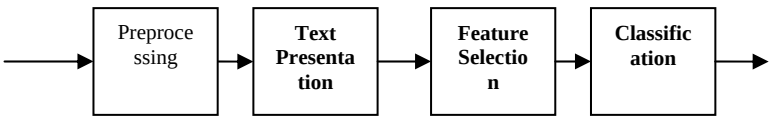


Figure 1. Text categorization process

In text classification, it has been proved that the term is the best unit for text representation and classification [1].Though a text document expresses vast range of information, unfortunately, it lacks the imposed structure of traditional database. Therefore, unstructured data, particularly free running text data has to be transformed into a

structured data. To do this, many preprocessing techniques are proposed in literature [2-6].After converting unstructured data to a structured form, we need an effective document representing model to build an efficient classification system.

In this paper, we proposed automatic text categorization method based on classical summarization techniques. The rest of this paper is organized as follows. Section 2 presents Text Categorization techniques. Section 3 describes the concepts of Text Mining and Information Extraction, Section 4 presents the Pyramidal model, Section 5 shows the organized file format, Section 6 explains the categorization based in feature terms, followed by results in section 7. Finally, we conclude in Section 8**.**

II. TEXT CATEGORIZATION TECHNIQUES

This section concerns the various Text Categorization approaches and previous research on text categorization. Sebastiani explained two kinds of approaches to text categorization in his research paper [12, 16].One is rule based class of approaches and the other is machine learning based approaches. Among the approaches, we consider three approaches: KNN (K Nearest Neighbor), NB (Naïve Bayes), and Decision Trees (J48) in this section, because of their popularity.

The first approach to text categorization is KNN. In k-nearest-neighbor classification, the training dataset is used to classify each member of a "target" dataset [20]. The structure of the data is that there is a classification variable and a number of additional predictor variables. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common amongst its *k* nearest neighbors (*k* is a positive integer, typically small). If $k = 1$, then the object is simply assigned to the class of its nearest neighbor. As k value goes up the computing time goes up but the advantage is that higher values of k reduces noise in the training data. Typically, k is in units or tens rather than in hundreds or thousands.

In 1992, KNN was applied to the classification of news articles by Massand. In 1999, Yang observed 12 approaches to text categorization with each other, and mentioned that KNN is one of recommendable approaches [15].In 2002, Sebatiani evaluated KNN as a simple and competitive algorithm with SVM which was evaluated as the best algorithm. Its disadvantage is that KNN costs very much time for classifying objects, given a large number of training examples because it must compute similarities of each unseen example for all individual training examples to select some of them [9].

Another popular approach to text categorization is Naïve Bayes (NB). Mitchell mentioned NB as a typical approach to text categorization [17]. The Naïve Bayes classifier is based on the Bayes rule of conditional probability. It makes use of all the attributes contained in the data, and analyses them individually as though they are equally important and independent of each other [20].

Its advantage is that it learns training examples with its higher speed than KNN classifier. However, its disadvantage is that an almost zero value of probability influences on the entire posteriori probability.

The final approach to text categorization is decision tree. This is a predictive machine-learning model that decides the target value of a new sample based on various attribute values of the available data [20]. The internal nodes of a decision tree denote the different attributes, the branches between the nodes gives the possible values that these attributes can have in the observed samples, while the terminal nodes tell us the final value (classification) of the dependent variable.

The attribute that is to be predicted is known as the dependent variable, since its value depends upon, or is decided by, the values of all the other attributes. The other attributes, which help in predicting the value of the dependent variable, are known as the independent variables in the dataset.

The J48 Decision tree classifier follows the simple algorithm. In order to classify a new item, it first needs to create a decision tree based on the attribute values of the available training data. Whenever it encounters a set of items (training set) it identifies the attribute that discriminates the various instances most clearly [20]. This feature that is able to tell us most about the data instances so that we can classify them the best is said to have the highest information gain.

For the other cases, we then look for another attribute that gives us the highest information gain. Hence we continue in this manner until we either get a clear decision of what combination of attributes gives us a particular target value, or we run out of attributes. In the event that we run out of attributes, or if we cannot get an unambiguous result from the available information, we assign this branch a target value that the majority of the items under this branch possess. In fact in several cases, it was seen that J48 Decision Trees had a higher accuracy than Naïve Bayes.

In this paper we proposed the method of Text Categorization on web documents using text mining and information extraction based on the classical summarization techniques. This method significantly improved the accuracy and degrees of relevancy.

III. TEXT MINING AND INFORMATION EXTRACTION

Text mining also called intelligent text analysis, text data mining, or knowledge discovery in text elucidates previously invisible patterns in existing resources by applying principles from several fields, such as computational linguistics, information retrieval, machine learning, and statistics [14]. Text mining is a specialization of data mining. It extracts useful patterns from unstructured text (books, articles, email messages, Web pages, etc.,), and converts it into a structured format. Here one major issue is sustainability. The user requirements, algorithms and languages are changing from time to time, thus without affecting the presently using applications we need to add special features to them.

Treating the Scalability and Execution time separately on each application can enable effective implementation of selected algorithm and platform.

Information Extraction is the main function in text mining which applies natural language processing (NLP) techniques to extract pieces of information (such as name, date, or affiliation), thus giving the document some structure [18]. But it becomes a complex task when dealing with files of multiple formats like *html* files, *pdfs*, *docs etc.*, as they involve various media and graphical items (Figure 2).

The web content needs to be parsed to useful text and then further conventions are to be made with various procedures. The aim of Knowledge Discovery is to "extract implicit, previously unknown, and potentially useful information from data" (Frawly, in his paper in 1991).Information Extraction mainly deals with identifying words or feature terms from within a textual file. Feature terms can be defined as those which are directly related to the domain [8].

In order to identify featured terms and process the data the web document has to undergo several operations which will be discussed in proceeding steps.

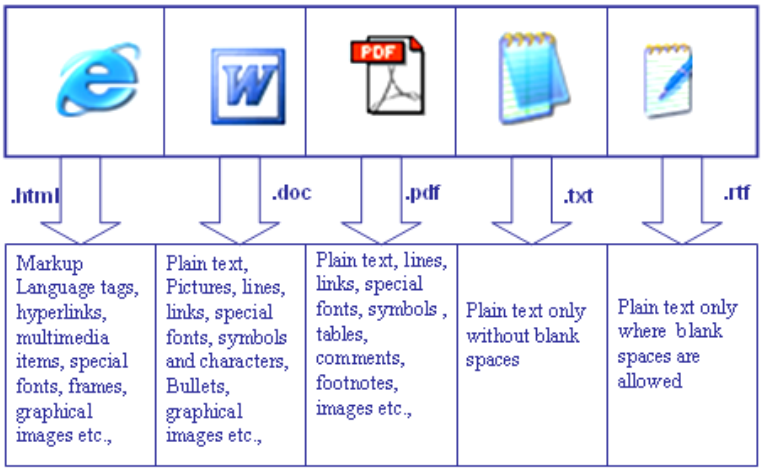


Figure 2. Representation of data

IV. PYRAMIDAL MODEL

If the data is trained before preprocessing it enables effective and reliable results with less time consumption. The basic steps needed during preprocessing must be recognized first so that unintentional and unrecognizable data can be eliminated in the very first step of preprocessing.

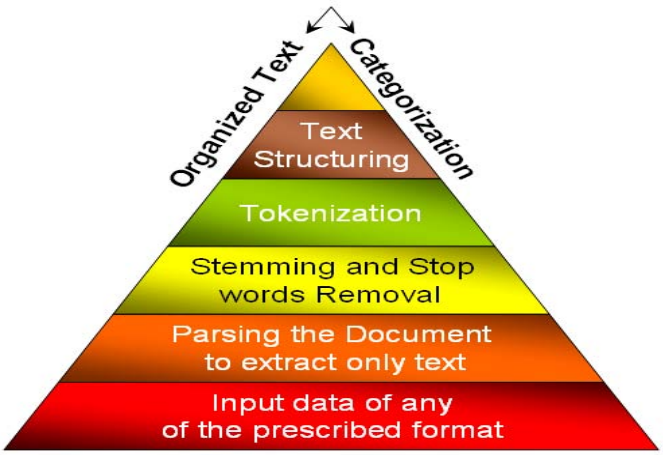
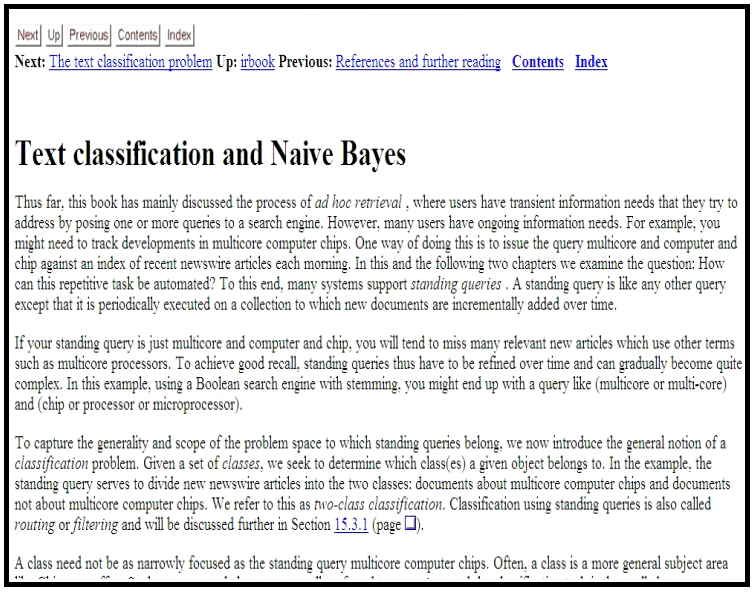


Figure 3. Pyramidai model

The operations involved in the text categorization are provided in the pyramidal model (Figure 3).

A. Parsing of web content

In the very first step, the web content is parsed and converted into text file removing all unneeded items like *html* tags, meta-data, comment information, images, bullets, buttons, graphics, links and all other hyper data [11]. The *html* test document (Figure 4) is parsed to extract only text. The parsed file is shown in Figure 5.



Next Up Previous Contents Index

Next: The text classification problem **Up:** irbook **Previous:** References and further reading **Contents** **Index**

Text classification and Naive Bayes

Thus far, this book has mainly discussed the process of *ad hoc retrieval* , where users have transient information needs that they try to address by posing one or more queries to a search engine. However, many users have ongoing information needs. For example, you might need to track developments in multicore computer chips. One way of doing this is to issue the query multicore and computer and chip against an index of recent newswire articles each morning. In this and the following two chapters we examine the question: How can this repetitive task be automated? To this end, many systems support *standing queries* . A standing query is like any other query except that it is periodically executed on a collection to which new documents are incrementally added over time.

If your standing query is just multicore and computer and chip, you will tend to miss many relevant new articles which use other terms such as multicore processors. To achieve good recall, standing queries thus have to be refined over time and can gradually become quite complex. In this example, using a Boolean search engine with stemming, you might end up with a query like (multicore or multi-core) and (chip or processor or microprocessor).

To capture the generality and scope of the problem space to which standing queries belong, we now introduce the general notion of a *classification* problem. Given a set of *classes*, we seek to determine which class(es) a given object belongs to. In the example, the standing query serves to divide new newswire articles into the two classes: documents about multicore computer chips and documents not about multicore computer chips. We refer to this as *two-class classification*. Classification using standing queries is also called *routing* or *filtering* and will be discussed further in Section 15.3.1 (page ❑).

A class need not be as narrowly focused as the standing query multicore computer chips. Often, a class is a more general subject area

Figure 4. *Html* file to be categorize

For html file the common rules used to remove html tags are:

1. If the sequence of bytes starting with “<!”, “</”, “<?”, then advance the pointer so that it points to the text after it is preceded by “!>”and “>” by simply skipping the bytes in between them.

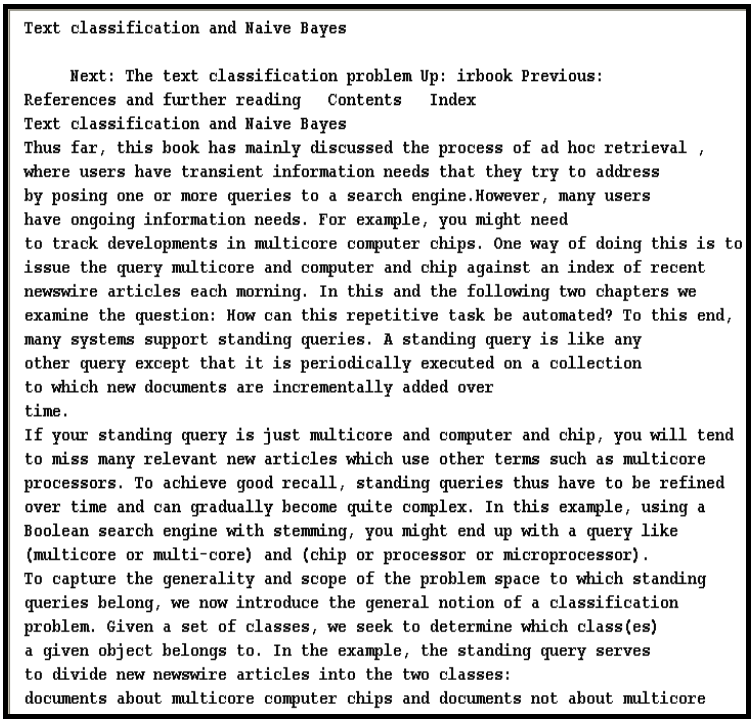
2. If the sequence of bytes starting with any of the attribute “<meta/http-equiv/content/char set” followed by space or slash, advance the position pointer so that it points at the next of “>”.

3. If the byte at position is one of ASCII TAB, ASCII LF, ASCII FF, ASCII CR, ASCII space then advance position to the next byte, then, repeat the rules.

4. All the html tags are treated in case insensitive manner.

The web document after parsing will be free unnecessary tags and the remaining text is to be processed further.

Many free tools are available on the web to parse *pdf* files, documents e*tc.*,



```
Text classification and Naive Bayes

     Next: The text classification problem Up: irbook Previous:
References and further reading   Contents   Index
Text classification and Naive Bayes
Thus far, this book has mainly discussed the process of ad hoc retrieval ,
where users have transient information needs that they try to address
by posing one or more queries to a search engine.However, many users
have ongoing information needs. For example, you might need
to track developments in multicore computer chips. One way of doing this is to
issue the query multicore and computer and chip against an index of recent
newswire articles each morning. In this and the following two chapters we
examine the question: How can this repetitive task be automated? To this end,
many systems support standing queries. A standing query is like any
other query except that it is periodically executed on a collection
to which new documents are incrementally added over
time.
If your standing query is just multicore and computer and chip, you will tend
to miss many relevant new articles which use other terms such as multicore
processors. To achieve good recall, standing queries thus have to be refined
over time and can gradually become quite complex. In this example, using a
Boolean search engine with stemming, you might end up with a query like
(multicore or multi-core) and (chip or processor or microprocessor).
To capture the generality and scope of the problem space to which standing
queries belong, we now introduce the general notion of a classification
problem. Given a set of classes, we seek to determine which class(es)
a given object belongs to. In the example, the standing query serves
to divide new newswire articles into the two classes:
documents about multicore computer chips and documents not about multicore
```

Figure 5. Parsed Text File

B. Stemming

Any text document, in general contains repetition of same word but with variations in the grammar such as word appearing to be in past, or in present tense and sometimes containing gerund (“ing” suffixed at the end).

Stemming refers to identifying the root of a certain word in the document [5, 6]. Stemming is of two types.

1) Derivational Stemming

2) Inflectional Stemming

Derivational stemming aims at creating a new word from an existing word, most often by changing the grammatical category.

e.g.: Ration*alize*- Ration*al*, Use*ful*-Us*e*

Music*al* – Mus*ic*, Fin*alize*-Fin*al*

Inflectional Stemming aims at confining normalized words to regular grammatical variants such as singular or plural or past or present.

E.g.: Classific*ation*-Classific,

Manag*ement*- Manage, Pay*ment*- Pay *etc.*

The two main advantages of stemming algorithms [7] are space efficiency and retrieval generality. The size of the inverted file can be reduced dramatically because many different words are indexed under the same stem and require only a single entry in the inverted file

C. Stop words removal

Some words are extremely common and occur in a large majority of documents. For example, articles such as “a”, “an”, “the”, “by” appear almost in every text but do not include much semantic information [11]. Since categorization is based on the featured terms not on commas, full stops, colons, etc., we remove them from document to list in tokens so that these words will not be stored in the signature file.

D. Feature Term Recognition

In order to recognize the terms relevant for categorization, it is essential to provide them with a knowledge base .A collective set of the entire feature terms is maintained in a Domain dictionary. In the proposed work the structure of the Domain dictionary consists of three levels in the hierarchy namely, Parent Category, Sub-category and word.

Parent categories define the main category under which any sub-category or word falls. A parent category is unique on its level in the hierarchy [13]. Sub-categories will belong to a certain parent category and each sub-category will consist of all the words associated with it (Table I).

TABLE I. THE PARENT AND SUB CATEGORIES

Parent Category	Sub category	Words
Text classification	Bayes	Naïve
	Regression	Network
		Neural
	Decision	Yes
		No
	Class	Classifier
	Text	..
	Document	..
	Categorize	..

E. Exclusion List

Unwanted data need to be removed from the processed file. For this purpose, a file consisting of a list of all adverbs, prepositions, articles are provided. If any similarity with the provided list occurs then it is removed from the base file.

V. CREATION OF ORGANIZED FILE

The text file obtained through the process of Information Extraction need to be converted into an organized file format. Comma Separated Variable (CSV) is an organized file format using which categorization can be done at word level.

For this we need to recognize the columns in the CSV file. The terms, term frequency and weight percentage are considered as columns in the CSV file. The algorithm pseudo code is shown in Table II.

TABLE II. ALGORITHM TO PRINT VALUES IN A CSV FILE THE PARENT

```
TFAndWeight (String temp, int  TotalTerms){
float count:=1.0;
int i=1;
While (i<TotalTerms)
If temp:=TermAt[i] then
Count:=Count+1;
else
continue;
float percent[i]= (Count / TotalTerms )X 100;
print(TermAt[i], count, percent[i]);
i++;
}
```

VI. CATEGORIZATION BASED ON THE FEATURE TERMS

Finally the corpus text is divided into words, delineated by white space and punctuation. All characters are lower-cased and stop words are removed. Then, the words are stemmed. The stemmed words are called terms. These terms are further referred as features [19].

Weights W (f, c) are now assigned by using different formulas, to the surviving features 'f' in category 'c'. We define W (f, c) as in equations 1 and 2.

$$w(f,c) = \frac{tf_{f.c}}{Ma\,x_c} \text{ ---------- (1)}$$

$$w(f,c) = tf_{f.c} \times idf_f \text{ ---------- (2)}$$

$$idf_f = \frac{T}{d_f}$$

Where $tf_{f.c}$ = the frequency of the feature f appearing in the category c,

T = the number of categories,

d_f = the number of categories that contain the feature f,

$Ma\,x_c$ = the maximum frequency of any feature in category c

The maximum weight percentage in the list is recognized and n topmost percentage terms are compared with the Domain Dictionary. The most significant terms in the sub category of a parent category are recognized. Thus the base category is recognized from the feature term selection. In the proposed method it can be observed that the entire categorization is based on term count rather than the percentage or terms.

VII. RESULT ANALYSIS

The resultant organized data file (CSV) is presented in the Figure 6 with term frequency and term percentage as attributes. The resultant category is tested on J48, Naïve Bayes and KNN categorization techniques on 5-fold cross validation and tabulated in Table III.

Text	count	percentag
String	Integer	double
text	1	20
classification	1	20
naive	1	20
bayes	1	20
some	2	12.5
text	1	6.25
classification	1	6.25
problem	1	6.25
up	1	6.25
irbook	1	6.25
previous	1	6.25
references	1	6.25
further	1	6.25
reading	1	6.25
next	1	6.25
contents	1	6.25
index	1	6.25
text	1	20
classification	1	20
naive	1	20
bayes	1	20
thus	1	5.26
far	1	5.26
this	1	5.26
book	1	5.26
has	1	5.26
mainly	1	5.26

```
Text,count,percentage
String,Integer,double
text,1,20
classification,1,20
naive,1,20
bayes,1,20
some,2,12.5
text,1,6.25
classification,1,6.25
problem,1,6.25
up,1,6.25
irbook,1,6.25
previous,1,6.25
references,1,6.25
further,1,6.25
reading,1,6.25
next,1,6.25
contents,1,6.25
index,1,6.25
text,1,20
classification,1,20
naive,1,20
bayes,1,20
thus,1,5.26
far,1,5.26
this,1,5.26
book,1,5.26
```

Figure 6. A CSV file in Excel sheet and in WordPad

The results from Table III indicate categorization based on "count" gives better performance than the other two attributes. Figure 7 shows the performance curves of J48, Naïve Bayes, and KNN categorization techniques. The success criteria for Text Categorization have significantly increased by using the proposed word level summarization techniques.

TABLE III. VARIOUS PARAAMETERS IN COMPARISON BETWEEN ATTRIBUTES

Classifier	Correctly classified instances in percent			Incorrectly classified instances in percent			Root Absolute Error root in percent			Root Relative Squared Error in percent		
	Term	Count	%	Term	Count	%	Term	Count	%	Term	Count	%
KNN	2	97	16	97	3	84	98	98	93	140	99	137
Naïve Bayes	2	97	28	98	3	81	98	97	98	140	95	100
J48	2	96	20	98	4	80	99	97	91	100	100	99

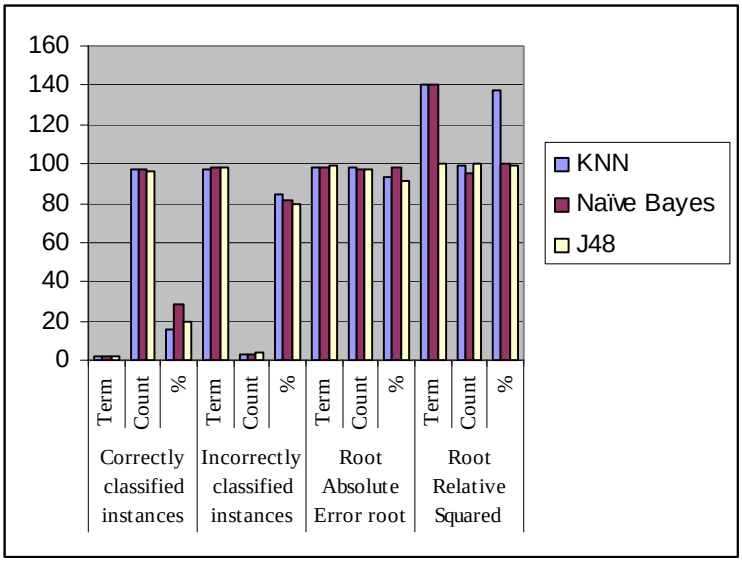


Figure 7. Performance curves

VIII. CONCLUSIONS AND FUTURE WORK

In this paper we proposed the method of Text Categorization on web documents using text mining and information extraction based on the classical summarization techniques. First web documents are preprocessed to establish an organized data file, by recognizing feature terms like term frequency count and weight percentage of each term. Experimental results shows, this approach of Text Categorization is more suitable for Informal English language based web content where there is vast amount of data built in informal terms. This method has significantly reduced the query response time, improved the accuracy and degrees of relevancy.

Future work includes the use of Meta information such as the structure of the document, patterns variations and evaluation for Text Categorization.

REFERENCES

[1] Lam, W. Ho, N.Y.(1998),Using a Generalized Instance set for Automatic Text Categorization SIGIR'98,pages 81-89.

[2] Dinesh, R., Harish, B. S., Guru, D.S., and Manjunath, S.2009. Concept of Status Matrix in Text Classification. In the Proceedings of Indian International Conference on Artificial Intelligence, Tumkur, India, pp. 2071 – 2079.

[3] Guru, D. S., Harish B. S., and Manjunath, S. 2009, "Clustering of Textual Data: A Brief Survey", In the Proceedings of International Conference on Signal and Image Processing, pp. 409 – 413.

[4] Mitra, V., Wang, C.J., and Banerjee, S. 2007. Text Classification: A least square support vector machine approach. Journal of Applied Soft Computing. vol. 7, pp. 908 – 914.

[5] Fung, G.P.C., Yu, J.X., Lu. H., and Yu, P.S. 2006. Text classification without negative example revisit. IEEE Transactions on Knowledge and Data Engineering. Vol. 18, pp. 23 – 47.

[6] Song, F., Liu, S., and Yang, J. 2005. A comparative study on text representation schemes in text categorization, Journal of Pattern Analysis Application, Vol 8, 2005, pp. 199 – 209.

[7] Porter, M.F. 1980. An algorithm for suffix stripping. Program, Vol. 14 (3), pp. 130 –137.

[8] Hotho, A., Nürnberger, A., and Paaß, G. 2005. A Brief Survey of Text Mining. Journal for Computational Linguistics and Language Technology. Vol. 20, pp.19 – 62.

[9] Salton, G., Wang, A., and Yang, C.S.1975. A Vector Space Model for Automatic Indexing. Communications of the ACM, Vol. 18, pp. 613 – 620.

[10] Bernotas, M., Karklius, K., Laurutis, R., and Slotkiene, A. 2007.The peculiarities of the text document representation, using ontology and tagging-based clustering technique. Journal of Information Technology and Control. Vol. 36, pp.217 – 220.

[11] Yang, Y., Slattery, S., and Ghani, R. 2002. A study of approaches to hypertext categorization. Journal of Intelligent Information Systems, Vol 18(2), pp. 219 – 241.

[12] Fabrizio Sebastiani, Machine Learning in Automated Text Categorization, Consiglio Nazionale delle, Italy.

[13] Atika Mustafa, Ali Akbar, "Knowledge Discovery using Text Mining: A Programmable Implementation on Information on Information Extraction and Categorization", International Journal of Multimedia and Ubiquitous Engineering, Vol.4, No.2.

[14] Juan José García Adeva and Rafael Calvo, "Miniing Text with Pimiento", University of Sydney, IEEE Internet computing, IEEE Computer Society, August, 2006.

[15] Yang Ymiing, Jan O. Pedersen, "A Comparative Study on Feature Selection in Text Categorization", In Proceedings of 14th International Conference on Machine Learning, 1997, pp. 412-420.

[16] F. Sebastiani, "Machine Learning Automated Text Categorization", *ACM Computing Survey*, Vol 34, No 1, pp1-47, January, 2002.

[17] T.M.Mitchell, *Machine Learning*, McGraw-Hill, 1997, ch 6.

[18] Mooney and Yong Nahm, "Text Mining with Information Extraction", *Multilingualism and Electronic Language Management:* Proceedings of the 4th International MIDP Colloquium,September 2003, Bloemfontein, South Africa, Daelemans, W., du Plessis, T., Snyman, C. and Teck, L. (Eds.) pp.141-160, Van Schaik Pub., South Africa, 2005

[19] Sue J. Ker, Jen-Nan Chen "A text categorization based on summarization technique" RANLPIR '00 Proceedings of the ACL-2000 workshop on Recent advances in natural language processing and information retrieval Association for Computational Linguistics - Volume 11

[20] www.d.umn.edu/~padhy005/Chapter5.html

AUTHORS PROFILE

Suneetha Manne is working as Assistant Professor in Information Technology at Velagapudi Ramakrishna Siddhartha Engineering College, Vijayawada, and ANDHRA PRADESH. She has received B.Tech, M.Tech Degree in Computer Science and Engineering and currently pursuing Ph.D at Osmania University. Her main research interest includes Text Mining, Knowledge engineering and Text summarization

Dr. S. Sameen Fatima M.S., Ph.D is working as Professor in Computer Science and Engineering Department, University College of Engineering, Osmania University, Hyderabad, Andhra Pradesh. She has received Ph.D in Computer Science and Engineering from Osmania University, Hyderabad. Hermain research interest includes Artificial Intelligence and Machine Learning.