Optimization of Text Summarization Based on Feature Selection and Classification

K.Gowri[1], Dr. R.Manicka Chezian[2]
[1](Research Scholar, Department of Computer Science, NGM College, Pollachi, India)
[2](Associate Professor, Department of Computer Science, NGM College, Pollachi, India)

Abstract: *Understanding the contents of a document via a text summarized version of the document requires a shorter time than reading the entire document, so that the summary text becomes very important. summarization requires a lot of time and cost when the documents are numerous and long document. Therefore, automatic summarization required to overcome the problem of reading time and cost. The propose features selection are the cornerstone in the generation process of the text summary. The summary quality is sensitive for those features in terms of how the sentences are scored based on the used features. The automatic text categorization, an ideal task-specific summary can be narrowly defined as the subset of most-informative features selected specifically with the categorization performance in mind. The propose system have three phase, first pre-processing document based on porter and Lancaster method to remove the unwanted words from document. The second method feature selection based on different type feature selection to weight each term. The Pruning techniques are also propose using ignore the feature based on TF and DF to further reduce the set of possible features words within a document prior to applying a method of feature selection. Finally classify the selected feature based on optimize navie bayes algorithm. The benchmark collections were chosen as the testbeds: Reuters-21578. The experimental result show better precision and recall compare with existing algorithms.*

Keywords: *Text summarization, pre-processing, Feature Selection, Text Classification*

I. INTRODUCTION

Text summarization is the problem of creating a short, accurate, and fluent summary of a longer text document. Summarization can also serve as an interesting reading comprehension test for machines. To summarize well, machine learning models need to be able to comprehend documents and distill the important information, tasks which are highly challenging for computers, especially as the length of a document increases.Text summarization is the process of producing shorter presentation of original content which covers no redundant and salient information extracted from a single or multiple document. A summary can be defined as a text that is produced from one or more texts, that contain a significant portion of the information in the original text(s), and that is no longer than half of the original text(s).

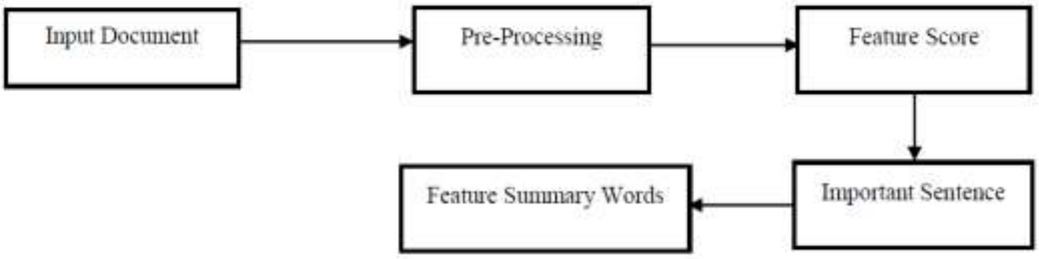

Fig. 1.1 General Process of Text Summarization

Automatic text summarization involves.

- Elimination of redundancy: The sentences in the text which convey the same meaning are said to be redundant and can be eliminated in the summary.
- Identification of Significant Sentences: Summary being a shorter representation of text requires including only salient sentences from the original document.
- Generation of Coherent Summaries: Sentences selected for summarization needs to be ordered and grouped so that coherence and readability is maintained.
- Metrics for Evaluating the Automatically Generated Summaries: In most of the cases the quality of the summary is judged by humans and hence automatic evaluation is a desirable feature.

There are two general approaches to automatic summarization: extraction and abstraction. Extractive methods work by selecting a subset of existing words, phrases, or sentences in the original text to form the summary. In contrast, abstractive methods build an internal semantic representation and then use natural language generation techniques to create a summary that is closer to what a human might express. Such a summary might include verbal innovations. Research to date has focused primarily on extractive methods, which are appropriate for image collection summarization and video summarization.

Text summaries can be either query relevant or generic summaries. Query relevant summaries contain sentences or passages from the document that are query specific. It is achieved by using conventional IR techniques. On the other hand, generic summary provides an overall sense of the document's content. In this method neither query nor any topic will be provided to summarizer. It is a big challenge for a summarizer to produce a good quality generic summary. In this paper, we propose an extractive technique for text summarization by using feature terms for calculating the relevance measure of sentences and extract the sentences of highest ranks. Then we perform their semantic analysis to identify semantically important sentences for creating a generic summary. Our proposed work generates a generic summary. There are various techniques that have been applied in text summarization. It includes

1. Statistical approach
2. knowledge-based approach
3. Linguistic Technique

Several methods to do automatic text summarization havebeen done, including the method that use techniqueslexical chains to obtain a text representation. The text summary using thetechniques to generate extraction path Bushy paragraph. The text summary using latentsemantic analysis (LSA), where the summary is based onthe semantic sentence. Text summarization has also beendone using genetic algorithms. Geneticalgorithm is used to find the optimal weights on thefeatures of text sentences.

II. RELATED WORK

Past literature that use the various summarization techniques are cited in this section. Most of the researchers concentrate on sentence extraction rather than generation for text summarization. The most widely used method for summarization is based on statistical features of the sentence which produce extractive summaries.

Automatic summarizers typically identify the most important sentences from an input document. Major approaches for determining the salient sentences in the text are term weighting approach [1], symbolic techniques based on discourse structure [2], semantic relations between words [3] and other specialized methods [4]. While most of the summarization efforts have focused on single documents, a few initial projects have shown promise in the summarization of multiple documents.The techniques for automatic extraction can be classified into two basic approaches [5]. The first approach is based on a set of rules to select the important sentences, and the second approach is based on a statistical analysis to extract the sentences with higher weight.

Cluster based methods measures relevance or similarity between each sentence in a document with that of sentences selected for summary. Summaries address onto different "themes" appearing in the documents, which is incorporated through clustering. Clustering based methods become essential to generate a meaningful summary. Documents are usually written such that they address different topics one after the other in an organized manner.Graph theoretic Approach representation is an extractive summarization model, which provides a method to identify themes in the document. Preprocessing steps, namely, stop word removal and stemming are done before, to obtain graphical view of the documents. Sentences in the documents form nodes of an undirected graph.

Singular Value Decomposition (SVD) [9] is a very powerful mathematical tool that can find principal orthogonal dimensions of multidimensional data. It has applications in many areas and is known by different names: Karhunen-Loeve Transform in image processing, Principal Component Analysis (PCA) in signal processes and Latent Semantic Analysis (LSA) in text processing. It gets this name LSA because SVD applied to document word matrices, groups documents that are semantically related to each other, even when they do not share common words. In automatic summarization, similarity metrics are used for centrality-based context selection and in identification of redundant contexts. In general, similarity measures are either corpus-based or knowledge-based. Both of them have been used in extractive summarization. Corpus-based measures use term frequencies observed in a corpus to relate contexts to each other, while knowledge-based methods utilize predefined semantic relations between terms obtained from lexical resources.

The selection procedure is to identify a set of sentences that contain important information. Three criteria are optimized when selecting the sentences: relevance, redundancy and length. Relevance determines the importance of the information contained in a summary with respect to the topics covered in the source documents or a query in case of query-focused summarization. Redundancy measures the information overlap between the sentences selected for the summary. Given a restricted summary length, summarization systems try to maximize the relevance while minimizing the redundancy. The task of content selection is to identify which sentences in the source documents are worth taking into a summary.

Redundancy is a major issue in multi-document summarization where several documents on the same topic may have a substantial information overlap. Then, the selection of the most relevant sentences will yield a set of sentences with redundant information. Extract that consists of relevant but very similar sentences is not good. The joint optimization of both relevancy and redundancy is a complex task because properties of individual sentences are dependent on other sentences included in the summary. Some of the earlier multi-document summarization approaches handle these optimizations separately.

Traditional evaluation studies typically rely on human subjects,either for creating the ideal summaries, or for judgingthe usefulness of different summaries. We propose a hybridapproach specifically targeting evaluation of the performanceof a summarization technique in automatic textcategorization. In the process, we do define an ideal summary,but instead of measuring an explicit agreement of anygiven summary with the ideal, we compare the categorizationperformance obtained with the actual and ideal summaries.Arguably, the proposed evaluation methodology isquite narrow and ignores other important aspects of a summary.

Recently, many researches handle the issue of the features selection (FS) process. Due to its importance, FS affects the quality of applications performance [6]. FS aims in identifying which features are important and can represent the data. In [7] the authors demonstrated that, embedding FS in a system may help effectively as follow. FS reduces the dimensionality, remove irrelevant data, and remove redundant features. Also, in hand of machine learning process, FS can reduce the amount of data which are needed. Consequently, it improves the quality of system results.

MapReduce framework is successfully utilized for a numbers of text processing taskssuch as stemming [8], distribute the storage and computation loads in a cluster [9],text clustering [10], information extraction [11], storing and fetching unstructured data[32], document similarity algorithm [12], natural language processing [13] and pairwisedocument similarity [14]. Summarizing large text collection is an interesting andchallenging problem in text analytics. A number of approaches are suggested for handlinglarge text for automatic text summarization [15, 16]. A MapReduce based distributedand parallel framework for summarizing large text is also presented.

III. PROPOSE METHODOLOGY

The propose feature selection process, each feature (term or single word) is assigned with a score according to a score-computing function. Then those with higher scores are selected. These mathematical definitions of the score-computing functions are often defined by some probabilities which are estimated by some statistic information in the documents across different categories. A number of feature selection metrics have been explored in text categorization, among which information gain (IG), chi-square (CHI), Mutual information (MI), Ng-Goh-Low (NGL), Galavotti-Sebastiani-Simi (GSS), odds ratios (OR) are considered most effective. To improved results are obtained by the proposed system in the text classification. Automatic text summarization approach to overcome the difficulties in the existing summarization approaches. Here, optimized Naïve Bayesian Classification approach is utilized to identify the necessary keywords from the text. Bayes method is machine learning method to estimate the distinguishing keyword features in a text and retrieves the keyword from the input based on this information. The features are generally independent and distributed. Scoring is estimated for the retrieved sentence to compute the word frequency. The combination of this Naïve Bayesian, scoring concept helps to improve the summarization accuracy.

1.1 Pre-Processing

Pre-processing is structured representation of the originalinputted text. The importance of pre-processing is used inalmost every developed system related with text processingand natural language processing. Pre-processing phaseincludes words identification, sentences identification,stop words elimination, language stemmer for nouns andproper names, allowing input in proper format andelimination of duplicate sentences or words.

1.1.1 Stop Words Elimination

Stop words are a division of natural language. The motive that stop-words should be removed from a text is that they make the text look heavier and less important for analysts. Removing stop words reduces the dimensionality of term space. The most common words in text documents are articles, prepositions, and pro-nouns, etc. that does not give the meaning of the documents. These words are treated as stop words. Example for stop words: the, in, a, an, with, etc.

1.1.2 Word Stemming

Porters stemming algorithmis one of the most popular stemming many modifications and enhancements have been made and suggested on the basic algorithm. It is based on the idea that the suffixes in the English language are mostly made up of grouping of smaller and simpler suffixes. It has five steps, and within each step, rules are applied until one of them passes the conditions. If a rule is accepted, the suffix is removed consequently, and the next step is performed. The resultant stem at the end of the fifth step is returned.

Removing suffixes by automatic means is an operation which is especiallyuseful in the field of information retrieval. In a typical IR environment,one has a collection of documents, each described by the words in thedocument title and possibly by words in the document abstract. Ignoring theissue of precisely where the words originate, we can say that a document isrepresented by a vector of words, or \terms\.

Paice/Husk Stemmer: The Paice/Husk Stemmer is a simple iterative Stemmer – thatis to say, it removes the endings from a word in an indefinitenumber of steps. The Stemmer uses a separate rule file, whichis first read into an array or list. This file is divided into aseries of sections, each section corresponding to a letter of thealphabet. The section for a given letter, say "e", contains therules for all endings ending with "e", the sections beingordered alphabetically. An index can thus be built, leadingfrom the last letter of the word to be stemmed to the first rulefor that letter.

When a word is to be processed, the stemmer takes its lastletter and uses the index to find the first rule for that letter. Therule is examined, and is accepted if:

- It specifies an ending which matches the last letters of the word.
- Any special conditions for that rule are satisfied (e.g., the so-called 'intact' condition, which ensures that the rule is only fired if no other rules have yet been applied to the word).
- Application of the rule would not result in a stem shorter than a specified length or without a vowel.

If a rule is accepted then it is applied to the word. If it is notaccepted, the rule index is incremented by one and the nextrule is tried. However, if the first letter of the next rule doesnot match with the last letter of the word, this implies that noending can be removed, and so the process terminates.

1.2 Feature Selection

1.2.1 X^2 Statistic

The X^2statistic measures the lack of independence between t and c. Using the two-way contingency

table of a term t and a category c, where A is the number of times t and c co-occur, B is the number of times the t occurs, and N is the total number of documents, the term-goodness measure is defined to be:

$$X^2(t,c) = \frac{N^*(AD - CB)^2}{(A+C)^*(B+D)^*(A+B)^*(C+D)} \qquad (1)$$

The x^2 statistic has a natural value of zero if t and c areindependent. We computed for each category the x^2 statisticbetween each unique term in a training corpus and thatcategory, and then combined the category-specific scores ofeach term into two scores:

$$X^2_{avg}(t) = \sum_{i=1}^{m} P_r(C_i) X^2(t, C_i) \qquad (2)$$

$$X^2_{max}(t) = max_{i=1}^{m}\{X^2(t, C_i)\} \qquad (3)$$

1.2.2 Mutual Information

MI, used to represent the correlation between twovariables (feature and category). A presents the number ofdocuments which belongs to cjand contains t, B presentsthe number of documents which doesn’t belong to cj butcontains t, C presents the number of documents whichbelongs to cj but doesn’t contain t, D presents the numberof documents which neither belongs to cj nor contains t.N=A+B+C+D, N presents the total number of documentsconcluded in the training set.

The MI between t and cj can be defined as:

$$MI(t, C_j) = log\frac{P(t\Lambda C_j)}{P(t)P(C_j)} = log\frac{P(t|C_j)}{P(t)} = log\frac{A \times N}{(A+C)(A+B)} \qquad (4)$$

1.2.3 Information gain

Information gain (IG) measures the amount ofinformation obtained for category prediction by knowingthe presence or absence of a term in a document. Let $m\ \{C_i\}_{i=1}^{m}$denote the set of categories in the target space. Theinformation gain of term t is defined to be:

$$G(t) = -\sum_{i=1}^{m} P(C_i) logP(C_i) + P(t)\sum_{i=1}^{m} P(C_i|t)\log P(C_i|t) + P(\bar{t})\sum_{i=1}^{m} P(C_i|\bar{t})\log\ P(C_i|\bar{t}) \qquad (5)$$

1.2.4 Odds ratio

Odd ratio is designed for two-class classifier, which isdefined as follow:

$$OR(t) = log_2 \frac{odds(t|pos)}{odd(t|neg)}$$

$$= log_2 \frac{P(t|pos)(1 - P(t|neg))}{P(t|neg)(1 - P(t|pos))} \qquad (6)$$

1.2.5 Correlation Coefficient

The Correlation Coefficient (CC) is a variant of X^2metric. The coefficient is reported to have better performance thanX^2". They say so, because CC "selects wordsthat correlate with c (i.e. are positive) and does notselect those words which correlate with $\bar{c}$, unlike theX^2 statistic". The NGL CC value can be computed as follows:

$$CC = \frac{\sqrt{N}.(AD - CB)}{\sqrt{(A + C)(B + D)(A + B)(C + D)}} \qquad (7)$$

1.2.6 Optimized $\boldsymbol{X^2\ and\ CC}$ statistic

They remove the $\sqrt{N}$ factor, and the denominator completely. They describe the $\sqrt{N}$ factoras being unnecessary. They also remove the denominator, $\sqrt{(A + C)(B + D)(A + B)(C + D)}$, by givingthe reason that the denominator gives high Correlation Coefficient score to rare words, and rare categories. The CC value can be computed asfollows:

$$ptimized = AD - CB \qquad (8)$$

1.3 Optimized Naïve Bayes classification

Naïve Bayes classification used to calculate the probability of a sentence s with k features like $F_1, F_2, \dots . F_k$be the set S or not? by the following formula

$$P(s \in S|F_1, F_2, \dots \dots F_k) = P(F_1, F_2 \dots . F_k|s \in S) \times \frac{P(s \in S)}{P(F_1, F_2 \dots . F_k)} \qquad (9)$$

Assuming that the feature is independent formula (9) converted to

$$P(s \in S|F_1, F_2 \dots . F_k) = \frac{\prod P(F_j|s \in S)P(s \in S)}{\prod P(F_j)} \qquad (10)$$

Using logarithmic rule (10) into:

$$P(s \in S|F_1, F_2 \dots . F_k) = \log\ (P(s) + \sum \log P(F_j|s) \qquad (11)$$

The system was able to learn from data. Some features used by their system include the presence of uppercase words, length of sentence, structure of phrase and position of words. The author assumed the following:

s = a certain sentence, S = the sentences in the summary, and $F_1, F_2 \ldots . F_k$the features

Feature extraction can be used for representing the important level of sentence. Some of them are thematic features; they are TF-IDF score, keyword extraction, key phrase extraction, similarity with title, inclusion of numerical, time, and entity data, and centrality. The thematic feature helps the reader more easily understand the document and provide additional information to help the reader comprehend the content. Then location features also used for consideration as part of features extraction. Sentence location and sentence relative length are features that defined by the location in the document.

IV. EXPERIMENTAL RESULTS

Even though these numbers are not comparable to other results since a subset and not the complete Reuters 21578 split was used, they provide still interesting Insights. Especially the fact, that for the same weighting function and the same dimensionality, it happens that, e.g., the breakeven value is higher compared to another function but the eleven-point precision is lower, compared to the same function. It also shows that" MSF" could be an interesting alternative to chi-square and information gain, not only for feature selection in text classification, but also to weight the importance of features in other classification tasks.

4.1 Precision-recall

The properties or the expected behaviors of text summarization systems can vary. For example, for one system it is better to return mostly correct answers, while in another it is better to cover more true positives. There is a trade-off between precision and recall: if a classifier says "True" to every category for every document, then it receives perfect recall, but very low precision. However, it can be easily seen that if a classifier says "False" for every category, except one which is correct (TP = 1, FP = 0) then it will have a precision equal to 1 but a very low recall. That is why it makes comparison between systems easier if the system is characterized by a single value, the breakeven point (BEP), which is the point at which precision equals recall. This can be achieved by tuning the parameters of the system. When there is no such point (because TP, FP and FN are natural numbers) the average of the nearest precision and recall is used, and is called interpolated BEP. For example, in ranking categorization models for each class an optimal τi CSV threshold has to be determined such that$P \cong R$. If $CSV_i\,(d_{j)} \geq \tau_i$then the classifier says "True", otherwise says "False".

4.2 11-point average precision

The 11-point average precision is another measure for representing performance with a single value. For every category the τi CSV threshold is repeatedly tuned such that allow the recall to take the values $0.0, 0.1, \ldots, 0.9, 1.0$. At every point the precision is calculated and at the end the average over these eleven values is returned. The retrieval system must support ranking policy.

The following detailed algorithm for the calculation of this value. The precision and recall values for a given document and a threshold is calculated as

1. For each document calculate the precision and recall at each position in the ranked list where a correct category is found.
2. For each interval between thresholds 0˙0ʾ I˙0ʾ . . . ʾ0˙∂ʾ I˙0use the highest precision value in that interval as the "representative" precision value at the left boundary of this interval.
3. For the recall threshold of 1.0 the "representative" precision is either the exact precision value if such point exists, or the precision value at the closest point in terms of recall. If the interval is empty we use the default precision value of 0.
4. Interpolation: At each of the above recall thresholds replace the "representative" precision using the highest score among the "representative" precision values at this threshold and the higher thresholds.
5. Per-interval averaging: Average per-document data points over all the test documents at each of the above recall thresholds respectively. This step results in 11 per-interval precision scores.
6. Global averaging: Average of the per-interval average precision scores to obtain a single-numbered performance average. The resulting value is called the 11-point average precision.

Table 1 COMPARISON OF P/R USING EXISTING WITH PROPOSED SYSTEM

Algorithms	Recall									
	0.1	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1**
Existing	0.61	0.45	0.41	0.38	0.35	0.31	0.28	0.26	0.25	0.24
KNN	0.85	0.72	0.64	0.57	0.46	0.40	0.38	0.31	0.29	0.25
NB	0.91	0.85	0.72	0.64	0.55	0.46	0.41	0.39	0.33	0.31

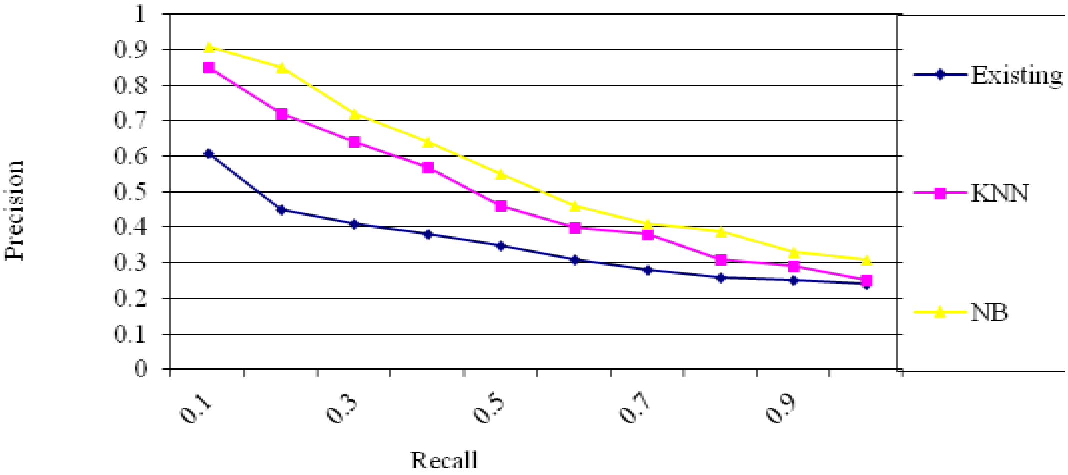

Fig. 1 Compare precision and recall

The fig .1 show the better result compare to existing classification algorithm optimization of naive bayes take less time to classify the document.

V. CONCLUSION

The propose features selection are the cornerstone in the generation process of the text summary. The summary quality is sensitive for those features in terms of how the sentences are scored based on the used features. The automatic text categorization, an ideal task-specific summary can be narrowly defined as the subset of most-informative features selected specifically with the categorization performance in mind. The propose system have three phase, first pre-processing document based on porter and Lancaster method to remove the unwanted words from document. The second method feature selection based on different type feature selection to weight each term. The Pruning techniques are also propose using ignore the feature based on TF and DF to further reduce the set of possible features words within a document prior to applying a method of feature selection. Finally classify the selected feature based on optimize navie bayes algorithm. The benchmark collections were chosen as the testbeds: Reuters-21578. The experimental result show better precision and recall compare with existing algorithms.

REFERENCES

[1] J. Salton and C. Buckley, “Term Weighting Approaches in Automatic Text Retrieval”, *Information Processing and Management*, vol. 24, no.5, pp.513-323, 1988.

[2] D. Marcu, “From Discourse Structures to Text Summaries”, *Proc. of the ACL 97/EACL-97 Workshop on intelligent scalable Text Summarization*, pp.82-88, Madrid, Spain, 1997.

[3] R. Barzilay and M. Elhadad, “Using Lexical Chains for Text Summarization”, *Proc. of the ACL Workshop on Intelligent Scalable Text summarization*, pp. 10-17, Madrid, Spain, 1997.

[4] D.R. Radev, H. Jing, and M. Budzikowska, “Centroid-based Summarization of Multiple Documents: Sentence Extraction, Utility-based Evaluation, and User Studies”, *Proc. of ANLP-NAACL Workshop on Summarization*, pp. 21-30, Seattle, Washington, April, 2000.

[5] C.Y. Lin and E. H. Hovy, “The Automated Acquisition of Topic signatures for Text Summarization”, *Proc. of the Computational Linguistics Conference*, pp. 495-501, Strasbourg, France, August, 2000

[6] H. Xingshi, Qingqing, Zhang, Na, Sun, Yan, Dong, "Feature Selection with Discrete Binary Differential Evolution*," in Artificial Intelligence and Computational Intelligence*, 2009. AICI '09. International Conference on, 2009, pp. 327-330.

[7] R. N. Khushaba, Al-Ani, A., Al-Jumaily, A., "Differential evolution-based feature subset selection," in Pattern Recognition, 2008. *ICPR 2008. 19th International Conference on*, 2008, pp. 1-4.

[8] Rajdho A, Biba M (2013) Plugging Text Processing and Mining in a Cloud Computing Framework. *In Internet of Things and Inter-cooperative Computational Technologies for Collective Intelligence Springer*, Berlin, Heidelberg, Germany, pp 369–390

[9] Balkir AS, Foster I, Rzhetsky A (2011) A Distributed Look-up Architecture for Text Mining Applications using MapReduce. *High Performance Computing, Networking, Storage and Analysis (SC),* 2011 International Conference. Seattle, US, pp 1–11

[10] Zongzhen H, Weina Z, Xiaojuan D (2013) A fuzzy approach to clustering of text documents based on MapReduce. *In Computational and Information Sciences (ICCIS), 2013 Fifth International Conference on IEEE*. Shiyang, China, pp 666–669

[11] Chen F, Hsu M (2013) A Performance Comparison of Parallel DBMSs and MapReduce on Large-Scale Text Analytics. *Proc. of the 16th International Conference on Extending Database Technology ACM.* New York, USA, pp 613–624

[12] Das TK, Kumar PM (2013) BIG Data Analytics: A Framework for Unstructured Data Analysis. *International Journal of Engineering and Technology (IJET)* 5(1):153–156

[13] Momtaz A, Amreen S (2012) Detecting Document Similarity in Large Document Collection using MapReduce and the Hadoop Framework.*BS Thesis. BRAC University*, Dhaka, Bangladesh, pp 1–54

[14] Lin J, Dyer C (2010) Data-Intensive Text Processing with MapReduce. *Morgan & Claypool Publishers* 3(1):1–177

[15] Elsayed T, Lin J, Oard DW (2008) Pairwise Document Similarity in Large Collections with MapReduce. *Proc. of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies*. Stroudsburg, US, pp 265–268

[16] Galgani F, Compton P, Hoffmann A (2012) Citation based summarization of legal texts. *Proc. of 12th Pacific Rim International Conference on Artificial Intelligence*. Kuching, Malaysia, pp 40–52.