Hash Based Document Retrieval by Utilizing Term Features

[1]Manisha Patel, [2]Umesh Lilhore

[1]M.Tech. Scholar, [2]Asst. Prof.
Computer Science and Engineering,
Bhopal, India

***Abstract*— As the digital data increases on servers different researcher have focused on this field. As various issues are arise on the server such as data handling, security, maintenance, etc. In this paper document retrieval was proposed that efficiently the fetch document as per query. Here hash based indexing of the dataset document was done by utilizing term features. In order to provide privacy for the terms each of this is identified by a unique number and each document has its hash index key for identification. Experiment was done on real and artificial dataset. Results shows that NDCG, precision, recall parameter of the work is better as compare to previous work on different size of datasets.**

***Index Terms*—Information Retrieval, Text Feature, Text Mining, Text Ontology.**

I. Introduction (Heading 1)

With the increase of digital text data on the servers. Text mining importance is increasing as this decrease lot of labor work for different use of text data. In this text mining research field classification of information and retrieval of documentation is highly required. So combination of various data mining techniques is done while gathering information from the text document [1]. As various researchers are working for improving accuracy of the work, but there is lot of improvement in the work for futher increasing the parameters.

As text data is highly unorganized because it contain natural language. So mining for retrieval of information from text data is crucial for the researcher. Different pre, post, processing steps are taken for improving the information quality. While in case of text document information retrieval, it is found that most of the document data is open for all. Due to this privacy of the text dataset is very low. So this work has focus on two issue first is text information retrieval and second is privacy maintenance of the dataset. Ways to mine the text and cluster the documents for better processing is our concern.

Even any small activity of human produces electronic data. For example, when any person buys a ticket online, his details are stored in the database. As most of electronic or digital data available on servers are in text form. This data is highly un-clustered or structure less but also suffered from the large amount of waste information. In this data good quality of information is also available for the scientific and industry purpose. As most of the historic data is available in text which need to be update but this required skilled labor or reader how have knowledge of the different terms for conversion. So considering all these facts in 1960 Pittsburg University has requirement of computer enabled system is desired which perform these task efficiently. In mid 1960 university has develop a computer enabled research assistant for performing the text reading [5]. In this computer programs Boolean logics were set with nearness expression in form of phrase were used.

So these program utilized full text query for retrieving document from the from the dataset. Here retrieval was done on the basis of content of the document not on the few set of keywords so this is term as Full Text Retrieval. So if the user want to fetch an document then this can be retrieve from the database based on the passed query. Here query is understand in form of terms and phrases. So this tends to find the FTR drawback where it do not understand the natural language of the user. But as FTR required less time for searching so this fact is overcome in terms of time efficiency. So text mining algorithms are applied to develop a document retrieval system which takes less execution time with high relevant data and protection against intruder for query as well as database.

II. Related Work

In [1] Text document clustering is used to group a set of documents based on the information it contains and to provide retrieval results when a user browses the internet. In this work results shows that proposed work has retrieve the text document efficiently by prior classification of the text files in the document. Here work has focus on reducing the dimension of the dataset. So dimension reduction is done in by two approaches first is reducing of noise or text which do not provide any information while second is removing of unwanted features from the document dataset.

K. Fragos et al. in [2] also concludes in favor of combining different approaches for text classification. The methods that authors have combined belong to same paradigm – probabilistic. Naïve Bayes and Maximum entropy classifiers are chosen to test on the applications where the individual performance is good. The merging operators are used above the individual results. Maximum and Harmonic mean operators have been used and the performance of combination is better than the individual classifiers.

Wild card approach used in [4] has implement the fuzzy approach where keyword set is used for matching the relevant text documents. Here comparison of these sets is done by the use of Edit Distance formula. So collection of text features in form of these keyword set is done in this work. This reduces the calculation overhead for the work, while storage of the work was also handled. Here privacy of the data is need to be maintained first for reducing the intruder activities.

In [7], the researcher has introduced dual indexing of the text document retrieval. Here first index consist of document index where text files are ranked based on there similarity. While in second index words are indexed where list of similar group words are list. Campus Net Search Engine (CNSE) is based on full-text hunt engine, but its not an complete text for indexing the document.

In [8] searching of document is done by using multi-keyword technique. Here frequent words are arranged in tree data structure as per there IDF value. For each term and document there is separate path, but number of path increases as number of documents is increasing. So for finding the new document recursive steps are required for other related documents. Recursion is require time, then comparison of word at each step is also time taken, which increases as the document in the dataset increases.

III. Proposed Methodology

As the mining is utilize in different type of data analysis so for the same all need to increase the different technique in the required area. So contributing the text mining is done in this work by the proposed method for retrieval of the document or articles in the group without having any prior knowledge of the documents. Whole work is explained in fig 1 and 2.

Preprocessing

Preprocessing is a process used for conversion of document into feature vector. Just like text categorizations the preprocessing also has controversy about its division [10].

Text preprocessing is consisting of words which are responsible for lowering the performance of learning models. Data preprocessing reduces the size of the input text documents significantly. It involves activities like sentence boundary determination, natural language specific stop word elimination. Stop-words are functional words which occur frequently in the language of the text (for example a, the, an, of etc. in English language), so that they are not useful for classification [3, 10, 13]. Let D is document [India is a great country. Its an country of different religion and caste.], Stop-word S is [a, are, an, and, am, for, is, its, when, where, etc.]. Then in pre-processing subtraction operation is done on these sets. Here D-S = [India, great, country, country, different, religion, caste].

```
Pre_processing(D, S)
Loop 1:x // x number of words in D
Loop 1:y // y number of words in S
If Not(D[x] – S[y])
PD←D[xs]
Endif
EndLoop
EndLoop
```

B. Feature Term

Term

The vector which contains the pre-processed data is use for collecting feature of that document. This is done by comparing the vector with vector KEY (collection of keywords) of the ontology of different area. So the refined vector will act as the feature vector for that document [11, 14].

So the list of words which are crossing the threshold are consider as the keywords or feature of that document.

$$[feature\ ] = \ mini\_threshold\ (\ [processed\_text]\ )\text{---}(1)$$

In this way term feature vector is created from the document.

C. Positive and Negative Feature set

Assign Term ID

Now assign number to each term of the different document. So that a dictionary of words with there number is created where each text is identified by separate number. Here words coming from different document which are already present in the dictionary is not updated. So those terms which are not present in the dictionary is insert in the dictionary with unique termed.

D. Document Hash Indexing

In this step document index is decide based on the terms collected from the document. Here all the term are arrange in decreasing order as per the terms frequency value of the terms in the document. So new order of the document term is 918465 this is based on the decreasing order of the term frequency. So from above table one has number is generate for selected document in similar fashion other document in the dataset get collected. Now as per the index value document is identify.

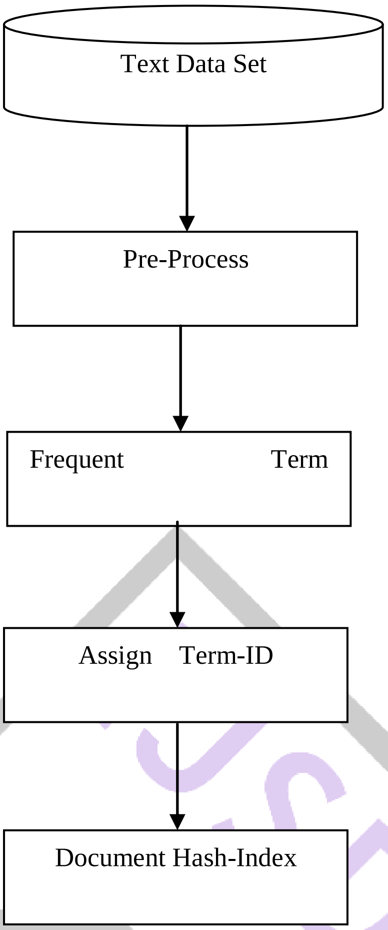

Fig.1 Block diagram of proposed Learning Model.

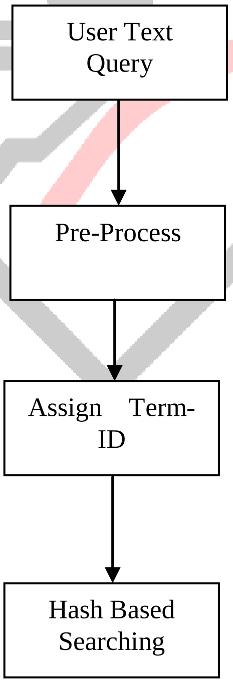

Fig.2 Block diagram of proposed Searching Model.

In this searching model whole step of pre-processing and assigning term-Id is same as done in previous steps, although term obtained after preprocessing is not filter as per there frequency in query. So hash based searching and retrieving related document is new for searching model. For convince words obtained from user text query is called as keywords.

E. Hash Based Searching

In this step as per the keywords (terms) from the user text query have their own term id while number are same as present in the dataset. Due to this term id privacy of the user query is increases. Now all term-id that are present in the text query act as key for the hash function where each document set from the matched index are retrieve. Now apply intersection between those sets. So top most common elements are fetch which act as similar documents.

Let term id be X which act as key and modulus function M is use for the hash base indexing. So output of the function is Y which is the index position of the insert key.

$$M(X) \rightarrow Y \text{ ------(2)// Modulus function}$$

$$M(X) = \|X, C\| \text{-------------------------(3)}$$

where C is fix constant use for finding the index position of the key.

Proposed Algorithm

Input: DD //Document Dataset, Dict //Stopword-Dictionary

Output: Hash_Table

```
Loop 1:n // n number of document in dataset
PD←Pre_processing(DD[n], Dict)
// Frequent Term
Loop 1:m // m number of keywords
C←Count(PD[m])
If C > T // T threshold
FT[n]←C
Endif
EndLoop
EndLoop
// Assign Term ID
Final_Term←Unique(FT)
Loop 1:u // u number of unique terms
TID←Assign(Final_term, FT)// TID Term ID
EndLoop
// Document Hash Table
Loop 1:n
SFT←Sort(FT[n]) // SFT Sorted frequent term
Hash_Table[SFT]←DD[n]
EndLoop
```

IV. Experiment and Results

B In order to implement above algorithm for document retrieval MATLAB 2012a tool was used. Here same work can be implement on other programming language as well. But as some of the function was inbuilt in the tool which help researcher to focus on the work. Experiment was done on real as well as on artificial dataset. Here different set of datset was use for retrieving documents.

Evaluation Parameter

As various techniques evolve different steps of working for classifying document into appropriate category. So it is highly required that proposed techniques or existing work need to be compare on same dataset. So following are some of the evaluation formula shown in equation number 4,5, 6and 7 which help to judge the classification techniques ranking.

Precision = (True_positive / (False_positive+ True_positive))-------(4)

Recall = (True_positive /(False_negative+ True_positive))-(5)

F-Measure = (2xPrecisionxRecall/ (Recall + Precision)) ---(6)

This work adopt NDCG [6, 12] as the performance evaluation measure. The NDCG measure is computed as

$$NDCG@P = Z_P \sum_{i=1}^{P} \frac{2^{l(i)}-1}{\log(i+1)} \quad (7)$$

where P is the considered depth, $l(i)$ is the relevance level of the i-th image and ZP is a normalization constant that is chosen to let the optimal ranking’s NDCG score to be 1.

Results

Table. 1. Comparison of accuracy value with previous work [15].

Comparison of Accuracy		
Query	Proposed Work	Previous work[15]
Q1	0.722222	0.666667
Q2	0.722222	0.611111
Q3	0.666667	0.388889
Q4	0.666667	0.555556

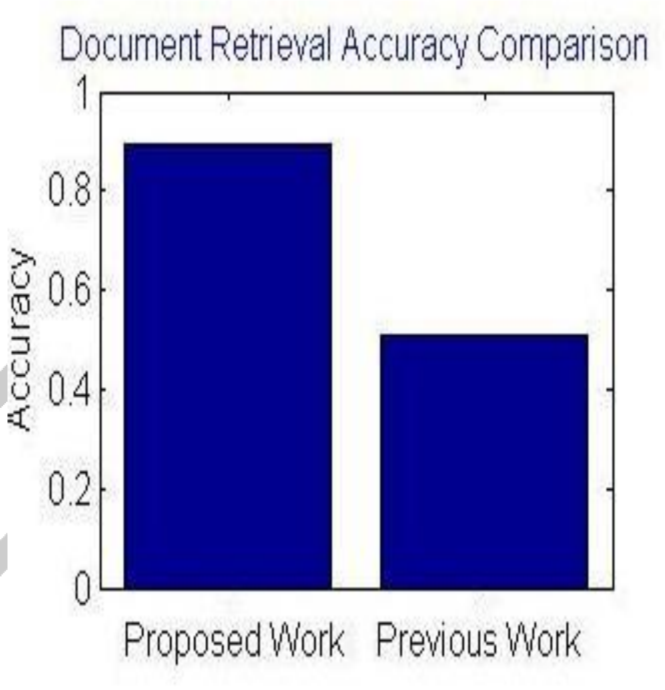

Fig. 3 Average accuracy values of proposed work comparison with previous work [15]

From above table 1and fig. 3 it is obtained that proposed work accuracy value is higher then previous work on different queries. As query set has good quality keywords results of proposed work is also high.

Table. 2. Comparison of precision value with previous work [15].

Comparison of precision values		
Query	Proposed Work	Previous work[15]
Q1	0.857143	0.714286
Q2	0.857143	0.714286
Q3	0.714286	0.428571
Q4	0.714286	0.714286

From above table 2 it is obtained that proposed work precision value is higher than previous work on different queries. As query set has good quality keywords results of proposed work is also high.

Table. 3. Comparison of Recall value with previous work [15].

Comparison of Recall		
Query	Proposed Work	Previous work[15]
Q1	0.6	0.555556
Q2	0.6	0.5
Q3	0.555556	0.3
Q4	0.555556	0.454545

From above table 3 it is obtained that proposed work recall value is higher then previous work on different queries. As query set has good quality keywords results of proposed work is also high.

Table. 4. Comparison of NDCG value with previous work [15].

Comparison of NDCG Values @7		
Query	Proposed Work	Previous work[15]
Q1	0.908375	0.775281
Q2	0.908375	0.810462
Q3	0.689134	0.484885
Q4	0.810462	0.79575

From above table 4 it is obtained that proposed work NDCG value is higher then previous work on different queries. As query set has good quality keywords results of proposed work is also high.

Table. 5. Comparison of execution time in second with previous work [15].

Comparison of execution time in second		
Query	Proposed Work	Previous work[15]
Q1	0.0260505	2.35873
Q2	0.0309973	2.36091
Q3	0.0156326	2.2921
Q4	0.0220363	3.17855

From above table 5 it is obtained that proposed work execution time value is comparatively low then previous work on different queries. As query set has good quality keywords results of proposed work is also high.

V. Conclusions

With the drastic increase of the digital text data on the servers, libraries it is important for researcher to work on it. Considering this fact work has focus on one of the issue of the document retrieval. Here many researchers has already done lot of work but that is focus only on the content classification where in this work document are classify. Proposed work has increase the retrieval efficiency of the work in all different evaluation parameters. So use of hash based indexing provide privacy with efficiency for document retrieval. As there is always work remaining in every because research is a never ending process, here one can implement similar thing for different other language.

REFERENCES

[1]. Aparna Humad, Vikas Solanki, A New Context Based Indexing In Search Engines Using Binary Search Tree, International Journal Of Latest Trends In Engineering And Technology (Ijltet) Vol. 4 Issue 1 May 2014.

[2]. Disputant Relation-Based Classification For Contrasting Opposing Views Of Contentious News Issues Souneil Park, Jungil Kim, Kyung Soon Lee, And Junehwa Song. Ieee Transactions On Knowledge And Data Engineering, Vol. 25, No. 12, December 2013.

[3]. Fabrizio Silvestri, Raffaele Perego And Salvatore Orlando. Assigning Document Identifiers To Enhance Compressibility Of Web Search Engines Indexes. In The Proceedings Of Sac, 2004.

[4]. Jian Ma, Wei Xu, Yong-Hong Sun, Efraim Turban, Shouyang Wang, And Ou Liu. “An Ontology-Based Text-Mining Method To Cluster Proposals For Research Project Selection”. Ieee Transactions On Systems, Man, And Cybernetics—Part A: Systems And Humans, Vol. 42, No. 3, May 2012

[5]. Jian Ma, Wei Xu, Yong-Hong Sun, Efraim Turban, Shouyang Wang, And Ou Liu. “An Ontology-Based Text-Mining Method To Cluster Proposals For Research Project Selection”. Ieee Transactions On Systems, Man, And Cybernetics—Part A: Systems And Humans, Vol. 42, No. 3, May 2012

[6]. K. Fragos, P.Belsis, And C. Skourlas, “Combining Probabilistic Classifiers For Text Classification”,Procedia - Social And Behavioral Sciences, Volume 147 Pages 307–312, 3rd International Conference On Integrated Information(Ic-Ininfo), Doi: 10.1016 /J.Sbspro .2014.07. 098 , 2014.

[7]. N. Cao, C. Wang, M. Li, K. Ren, And W. Lou, “Privacy-Preserving Multikeyword Ranked Search Over Encrypted Cloud Data,” Proc. Ieee Infocom, Pp. 829-837, Apr, 2014.

[8]. N. Cao, S. Yu, Z. Yang, W. Lou, And Y. Hou, “Lt Codes-Based Secure And Reliable Cloud Storage Service,” Proc. Ieee Info- Com, Pp. 693-701, 2012.

[9]. Oren Zamir And Oren Etzioni. Web Document Clustering: A Feasibility Demonstration. In The Proceedings Of Sigir, 1998.

[10]. Privacy Preserving Ranked Keyword Search Over Encrypted Cloud Data Dinesh Nepolean, I.Karthik, Mu.Preethi, Rahul Goyal And M.K. Vanethi. Volume 4, No. 11, November 2013.

[11]. Privacy-Preserving Multi-Keyword Ranked Search Over Encrypted Cloud Data Ning Cao,Cong Wang, Ming Li, Member, And Wenjing Lou, Ieee Transaction Parallel And Distributed Ssystems, Vol. 25, No. 1, January 2014

[12]. S. Keretna, C. P. Lim And D. Creighton, “Classification Ensemble To Improve Medical Named Entity Recognition”, 2014 Ieee International Conference On Systems, Man, And Cybernetics, San Diego, Ca, Usa, 2014.

[13]. S.Ramasundaram, “Ngramssa Algorithm For Text Categorization”, International Journal Of Information Technology & Computer Science (Ijitcs), Volume 13, Issue No : 1, Pp.36-44, 2014.

[14]. Yuefeng Li, Abdulmohsen Algarni, Mubarak Albathan, Yan Shen, And Moch Arif Bijaksana. “Relevance Feature Discovery For Text Mining”. Ieee Transactions On Knowledge.

[15]. Wenhai Sun, Bing Wang, Ning Cao, Ming Li, Wenjing Lou, Y. Thomas Hou And Hui Li . “Verifiable Privacy-Preserving Multi-Keyword Text Search In The Cloud Supporting Similarity-Based Ranking”. Ieee Transactions On Parallel And Distributed Systems, Vol. 25, No. 11, November 2014.