An Efficient Keyword Searching Algorithm for Information Retrieval from Desktop

S. Vijayarani, PhD
Assistant Professor
Department of Computer Science
Bharathiar University
Coimbatore

R. Janani, PhD
Coimbatore

S. Dinesh
Coimbatore

ABSTRACT
The growth of huge volume of text documents in the internet has lead the users to download and store the lot of information on their computers. hence, the retrieval of specific information from huge volume of documents is a challenging task. The main objective of this research work is to develop a tool which is used to perform the search process and retrieve the relevant information based on the query which is given by the user.The significant steps of this tool are, Document Collection, Searching and Retrieval. The documents (.txt, .pdf, .docx) are collected from the system (various folders), there searching task is carried out, after the Query keyword from the user. Now the tool will search the collected documents ay analyzing whether the given search query is found in the documents or not. This can be performed by using the existing and proposed algorithms. Normal search and Indexed search are existing algorithms and the Graph Based Keyword Search (GBKeyS) algorithm is a proposed algorithm. From the experimental result it is found that, the proposed algorithm produced the better results than existing searching algorithms.

Keywords
Text Mining, Information Retrieval, Keyword Search, Desktop Search, Normal Search Indexed Search, Graph based Search, PageRank

1. INTRODUCTION
Information Retrieval (IR) systems retrieves the documents that suit the demand of a user queries. The most well-known information retrieval systems are search engines such as Google, which find all documents applicable to a collection of provided terms on the World Wide Web. Text is known to consist of two fundamental units, namely the text and the term. Keyword Extraction is the automatic discovery of words which best describe a document's subject matter. Key phrases, key terms, key categories or just keywords or terminology used to define the terms describing the most relevant information found in the document. As documents are viewed as a bag of words, they can be represented by a vector model which can then be used as an input to the techniques described above, such as classifications, clustering, but this is not used for this method.

The documents are first converted into structured databases in information extraction, on which data mining techniques can be used to extract knowledge or interesting patterns. The outcome would be a prototype where it easily identifies all the individuals and their relationships with each other.

2. RELATED WORKS
[1] the techniques and applications for the text mining where the author to discuss the basic techniques, information retrials information extraction, topic tracking, Summarization, Categorization, clustering, information visualization for text mining. The application for the text mining is Publishing and media, Telecommunications, energy and other services Industries, Information technology sector and Internet, Banks, insurance and financial markets, Political institutions, political analysts, public administration and legal documents, Pharmaceutical and research companies and healthcare.

[2] discussed the text mining applications and issues of the text mining. The text mining areas are information extraction, information Retrieval, data mining, Natural Language Processing. The basic process of the text mining is Text Cleanup, Tokenization, Part of Speech Tagging, Text Transformation (Attribute Generation). Where the issues are occurred on the based on the application.

The support vector machine (SVM) is a training algorithm for learning classification and regression rules from data. The closet is an interesting substitute, proposed by pas quire et al. instead of mining the complete set of frequent item sets and their associations, association mining only needs to end often closed item sets and their equivalent rules, Tf–idf, short for term rate of occurrence–inverse document rate of occurrence, is a numerical statistic that is intended to reflect how important a word is to a document in a gathering or collection. It is also used as a weighting factor in data regain and text mining**[3].**

[4]deliberated the process of the text mining, Document Gathering, document pro processing, Text Transformation, Attribute Selection, and Pattern Selection. Techniques are Information Extraction, Information Retrieval, Natural Language Processing, Categorization, and Clustering. Applications are Security, Biomedical, Company Resource Planning, Market Analysis, and Customer Relationship Management. Issues are Intermediate Form, Multilingual Text Refining, and Domain Knowledge Integration.

{5] considered about the text mining classification, clustering and Extraction Techniques. The text classifications are Naive Bayes Classifier, Nearest Neighbor Classifier, Decision Tree classifiers, Support Vector Machines. The cluster methods are

Hierarchical Clustering algorithms, k-means clustering. The documents extraction is Named Entity Recognition, Hidden Markov Models, and Relation Extraction**[6].**

3. METHODOLOGY
The main objective of this research work is to develop a tool which is used to perform the search process and retrieve the relevant information based on the query which is given by the user. The significant steps of this research are, Document Collection, Searching and Retrieval. The documents (.txt, .pdf, .docx) are collected from the system (various folders), there searching task is carried out, after the Query keyword from the

user. Now the tool will search the collected documents ay analyzing whether the given search query is found in the documents or not.

3.1 Keyword Extraction

Information retrieval (IR) determines the documents of an unstructured nature that satisfies an information need from a document collection [7]. This system generally searches in collections of unstructured or semi-structured documents. The main applications of information retrieval systems are digital libraries, media search, search engine like desktop search, mobile search, and web search etc., this research work mainly focused on the desktop search to retrieve the file name based on user given keyword. Keyword extraction is tasked with the automatic identification of a collection of terms that best describe the subject of a document [8].

3.1.1 Normal Search Algorithm

In this research work the normal search verifies the keyword which exists in the particular document or not. Typically, a simple function is applied to the key to determine its place in the dictionary[9]. The instance S_i is taken one by one and each token T_m in S_i is compared with query given by the user. The string $S_i(T_m)$ goes to the storage buffer and it searches the keyword K_i. If $S_i=K_i \in D_n$ then the particular Si is labeled with the relevant class. If the keyword is not found, then the instance is considered as the outliers O_n. There are two types of cases namely best case and worst case are considered for normal search [10].

- Best case: The best case occurs when the search term is found in any one of the documents from D1, D2,….D_n
- Worst case: The worst case occurs when the search term is not found in any kind of documents from D1, D2,….D_n.

Algorithm 1: Normal Search Algorithm

Input : Documents D1, D2 ….Dn, Keyword W_i
Output : Documents which contains the keyword
Step 1 : K=0;
Step 2 : for (I-1- n);
Step 3 : Consider the documents one by one;
Step 4 : for (r=1 to m) {
Step 5 : verify if (W_i) is in D_j)
{
Retrieve the file into S_j
Go to Step 4
}
}
Else
Assign a class label, into S_j
K = K + 1;
Message: No Matches Found
Step 6: Stop the Process

3.1.2 Indexed Search Algorithm

Indexed Searching Algorithm indicate each array element with a particular index value and has that index value to the function in the form of M mod N (M is the index value and N is the user defined number) [11]. In this method, the list of significant keywords of various disciplines are stored in a single document. The information S_i is compared with the indexed documents which consists of the words arranging from W_1, W_2,…W_n. If Si is not found in this indexed searching document, then the normal search method is used D_1, D_2,…..D_n and searches for the keyword. If $S_i \in D_n$, then the information in the table is grouped into their relevant classes. If the normal search process fails to do this, then the instance is considered as outlier [12].

Algorithm 2: Indexed Search Algorithm

Input: Documents D_1, D_2……. D_n,Keyword W_i
Output: File which contains the given keyword

Step 1: K=0;
Step 2: for (I = 1 to n)
Step 3: Consider the Source title one by one S_i

For (j=I to MAX_entry {
Step 4: Verify if (W_i in Ind_Tab_j)
then consider its relevant document code and retrieve D_i }
Else {
Step 5: for (j = 1 to n) {
Step 6: Verify if (W_i in D_j) {
Assign a document into S_i

Go to Step 2}

K=K+1} }

Step 7: Stop the Process

3.1.3 Graph Based Keyword Search (GBKeyS)Algorithm

The GBKeyS algorithm is the graph based search algorithm for searching the particular word in the collection of documents. This algorithm consists two important steps such as topic learning and keyword extraction. The inputs are the documents DC which consists of document D_i and its respective manual labeled keyword W_i (D_i, $K_i \in$ DC) [13]. Another corpus is testing corpora DT consisting different documents. w is a variable controlling the window size while creating the word graph. The topic model controls the topic number of the result using variable K [14].

First step is to find out all the word pairs in Di, and then these pairs are filtered by the learning algorithm LK. The remaining words pairs shows a semantic similarity and a new edge is added between the two words in each pairs. The normal PageRank algorithm to rank the nodes in graph G and obtain ranked words RW. Then the sorting is done for the ranked words according its ranking and choose Top-N words to become the keywords of Di

Algorithm 3: GBKeyS Algorithm

Input: Topic learning corpora DC; Test corpora DT
Window Size w; Topic number K; The length S of each topic set.
Step 1: Initialize the graph with keywords Ki ∈ DC
Step 2: LK ← GRAPH(DC, K)
Step 3: for each document Di ∈DT :
Step 4: do
Step 5: G ←− graph(Di, w)
Step 6: for (t1, t2) in all two-tuples terms of document:
Step 7: do if (t1, t2) ∈ LK
Step 8: do G.addEdge(t1, t2)
Step 9: End IF
Step 10: End For
Step 11: RW ←− PageRank(G)
Step 12: T opN ←− sort(RW, N)
Step 13: End For

4. RESULTS AND DISCUSSION

Performance measurement is generally defined as regular measurement of outcomes and results, which generates reliable data on the effectiveness and efficiency of programs. The performance measure in this project work is used to identify the best method for retrieving the information. In order to measure the performance of the proposed search methods four different criteria are used; they are correctly retrieved instances, incorrectly retrieved instances, time taken for searching and accuracy.

Table 1: Confusion Matrix

	A Document which belongs to the particular category	A Document which does not belong to the particular category
Document category accepted by the classifier	TP	FP
Document category rejected by the classifier	FN	TN

This matrix is established in the terms,

- True Positives (TP) – It is defined; the similar documents are classified in the same category.
- True Negatives (TN) – It is defined; the dissimilar documents are classified in the different category.
- False Positives (FP) – It is defined; the dissimilar documents are classified in the same category.
- False Negatives (FN) – It is defined; the similar documents are classified in the different category.

$$\text{Correctly Retrieved} = \frac{TP}{TP+FN} \text{.....(Eq.1)}$$

$$\text{Incorrectly Retrieved} = \frac{TN}{TN+FP} \text{.....(Eq.2)}$$

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \text{.....(Eq.3)}$$

***Search Time**:*

The time complexity of an algorithm quantifies the amount of time taken by an algorithm to run as a function of the length of the string representing the input. Time complexity is commonly estimated by counting the number of elementary operations performed by the algorithm, where an elementary operation takes a fixed amount of time to perform. Thus the amount of time taken and the number of elementary operations performed by the algorithm differ by at most a constant factor. Search time measures the amount of time required for searching and the information.

All the experiments are carried out on a 2.00 GHz Intel CPU with 1 GB of memory and running on windows 10. This project implemented the algorithm to achieve the accurate categories of documents and verified the success of text classification.

Table 1 shows the performance of existing and prosed algorithms. From this inferred that, the GBKeyS algorithm performs well when compared to existing search algorithm. The GBKeyS algorithm retrieves the documents with higher accuracy.

Table 2: Performance Analysis

Algorithm	Correctly Retrieved (%)	Incorrectly Retrieved (%)
Normal Search	89.26	10.74
Indexed Search	91.50	8.50
GBKeyS	96.68	3.32

The comparison of correctly and incorrectly documents are shown in Figure 1. From this, the normal search algorithm retrieves very less percentage of documents when compared to the indexed search. The GBKeyS algorithm outperforms when the number of documents are large.

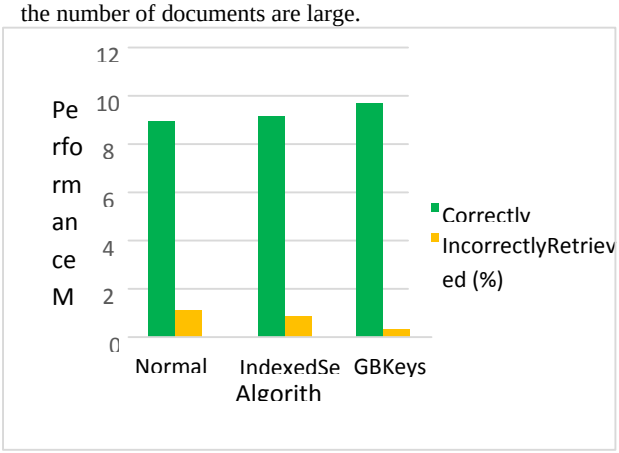

Fig 1. Performance Measure

The search time for given keyword comparison is shown in Table 2. The proposed algorithm searches the keyword with less time when compared to the existing techniques. This algorithm searches the query keyword given by the user within all the documents from personal computer.

Table 3: Search Time Comparison

Algorithm	Search Time (ms)
Normal Search	1523
Indexed Search	1477
GBKeyS	1186

Figure 2 represents the search time comparison of existing and proposed algorithms. From this, the proposed keyword searching algorithm gives the better accuracy.

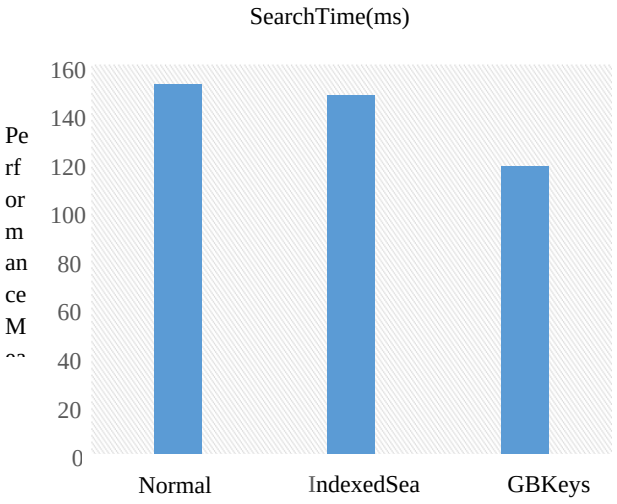

Fig 2. Keyword Search Time

The keyword searching algorithm accuracy is shown in Table 3. From this, the proposed algorithm yields the better accuracy when compared to existing algorithms. The proposed algorithm retrieves the documents based on the keyword with high accuracy.

Table 4: Accuracy of Keyword Searching Algorithm

Algorithm	**Accuracy (%)**
Normal Search	83.27
Indexed Search	92.88
GBKeyS	98.17

Figure 3 shows the accuracy comparison of existing and proposed systems. Based on the accuracy, the proposed algorithm retrieves the relevant documents based on the query.

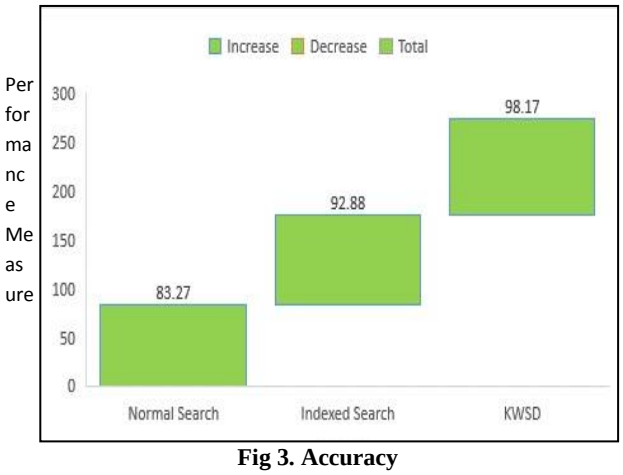

Fig 3. Accuracy

5. CONCLUSION AND FUTURE WORK

Keywords provide a compact representation of a document's content. There are many data mining algorithms that have been proposed for mining useful patterns from documents in preceding research. It seems that the discovered knowledge in the field of text mining is difficult and ineffective. In the existing system, the text has been extracted but it produced lower accuracy, precision and recall performance. Graph-based methods for keyword extraction are inherently unsupervised, and have fundamental aim to build a network of words (phrases) and then rank the nodes exploiting the centrality motivated measures. The main aim of this research work to develop a tool which is used to perform the search and retrieve the relevant information based on the query which is given by the user. The documents (.txt, .pdf, .docx) are collected from the system (various folders). Searching task is carried out, after getting the query keyword from the user. Now the tool will search the collected documents ay analyzing whether the given search query is found in the documents or not. This can be performed by using the existing and proposed algorithms. Normal search and Indexed search are existing algorithms and the Graph Based Keyword Search (GBKeyS) algorithm is the proposed one. The proposed algorithm, produced the better results during the retrieval step, the tool displays the documents which contains the query.

In future, this work to be enhanced with retrieving the sentences which is related from the query given by the user. To select the features of the documents the optimization techniques to be used. Then the documents to be classified based on its contents.

6. REFERENCES

[1] Vishal Gupta, Gurpreet S. Lehal,"A Survey of Text Mining Techniques and Applications", journal of emerging technologies in web intelligence, august 2009

[2] Mrs.B.Meena Preethi, Dr.P.Radha, "A Survey Paper on Text Mining - Techniques, Applications And Issues", IOSR Journal of Computer Engineering

[3] Gaikwad Varsha R, Patil HarshadaR,"Survey Paper on Pattern Discovery Text Mining for Document Classification", International Journal of Computer Applications (0975 – 8887) Volume 112 – No 12, February 2015

[4] R. Janani, Dr. S. Vijayarani, "Text Mining Research: A Survey", International Journal of Innovative Research in Computer and Communication Engineering, Vol.4, Issue 4, April 2016

[5] G. Salton, C. S. Yang, C. T. Yu, ―A Theory of Term Importance in Automatic Text Analysis‖, Journal of the American society for Information Science, 26(1), 33-44, 1975.

[6] J. D. Cohen, ―Highlights: Language and Domainindependent Automatic Indexing Terms for Abstracting Journal of the American Society for Information Science, 46(3): 162-174, 1995

[7] M. Ortuño et al., ―Keyword detection in natural languages and DNA‖, Europhys. Lett. 57, 759, 2002

[8] J.P. Herrera, P.A. Pury, ―Statistical keyword detection in literary corpora‖, The European physical journal, 2008

[9] Turney P. D., ―Learning algorithms for keyphrase extraction‖, Information Retrieval, 2: pp 303-336, 2000

[10] Hulth A. ―Improved automatic keyword extraction given more linguistic knowledge‖, Proceedings of the 2003 conference on Empirical methods in natural language processing, pp. 216-223. Association for Computational Linguistics, Morristown, NJ, USA, 2003

[11] Turney P., ―Coherent Keyphrase Extraction via Web Mining‖, Proceedings of the Eighteenth International Joint

Conference on Artificial Intelligence (IJCAI-03), pp. 434-439, 2003

[12] Tang J. et al.: Loss Minimization Based Keyword Distillation, Lecture Notes in Computer Science Volume 3007, pp 572-577, 2004

[13] Yasin Uzun, "Keyword Extraction Using Naïve Bayes", Bilkent University, Computer Science Dept., Turkey, 2005

[14] Medelyan O., Witten H. ‖Thesaurus based automatic keyphrase indexing‖, Proceedings of the 6th ACM/IEEECS joint conference on Digital libraries, Pages 296-297, 2006