# SYSTEM FOR DOCUMENT SUMMARIZATION USING GRAPHS IN TEXT MINING

Prashant D. Joshi[1], M. S. Bewoor[2], S. H. Patil[3]

[1]Deptt. of Computer Engineering, Researcher, Bharati Vidyapeeth University, Pune, India.
[2]Deptt. of Computer Engg., Asst. Professor, Bharati Vidyapeeth University, Pune, India.
[3]Deptt. of Computer Engineering, Professor, Bharati vidyapeeth University, Pune, India.

## *ABSTRACT*

*Summarization of text documents is increasingly important with the amount of data available on the Internet. The large majority of current approaches view documents as linear sequences of words and create query-independent summaries. However, ignoring the structure of the document degrades the quality of summaries. Furthermore, the popularity of web search engines requires query specific summaries. Here one method is used to create query-specific summaries by adding structure to documents by extracting associations between their fragments. This paper has practically implemented graph method for text mining and accordingly documents summary is generated. This system is developed using java programming language and ORACLE. Here text files are stored at particular drive and document graph of each text file is generated by using IR ranking algorithms. When input query is entered it is checked on document graph using summarization algorithm. Summary of matching text files will be displayed as an output. Various results are taken and this system can be implemented in desktop, network environment for accessing files within a short period of time .further new algorithm like top 1 expanding search algorithm can be added to improve the performance of this system.*

***KEYWORDS:*** *Query, Document summarization, Document Graph structure.*

## I. INTRODUCTION

Due to rapid growth of Electronic document there is a need for effective search algorithm.WWW contains so many electronic document and user want to retrieve it within short period of time. Text search and Document summarization are two essential technologies that complement each other. The importance of data/text mining and knowledge discovery is increasing in different areas like: telecommunication, credit card services, sales and marketing etc. Text mining is used to gather meaningful information from text and it includes tasks like Text Categorization, Text Clustering, Text Analysis and Document Summarization. Text Mining examines unstructured textual information in an attempt to discover structure and implicit meanings within the text.

This paper is mainly concentrating on the Document summarization for text mining. Summarization techniques will help us to reduce our access time while we are retrieving data from the internet. The summarization concept is mainly started on the principle of index of books. In books when person want to search particular topic he or she will refer index of book and then that point will be retrieved in the less time. Traditionally Query Summarization is based on the BOW (Bag of Words) approach, in which both the query and sentences are represented with word vectors. [3], this approach suffers from the shortcoming that it merely considers lexical elements (words) in the documents, and ignores semantic relations among sentences. Second way is Natural Language processing where input query is processed by reading each word of the file and then display paragraph which is matching with input query [1]. This is very difficult for the file which has huge amount of data. Text Summarization is the process of identifying the most salient information in a document. [4].In this paper we are referring Document Graph based summarization method for information retrieval and displaying the summary of that text file.

# II. RELATED WORK

## 2.1 Query Summarization Characteristic

**1.** The sentences included in the summary are required to be closely relevant to the query.

**2.** The performance of Query Summarization relies highly on accurate measurement of text similarity.

**3.** In Multi document summarization more than one document are present with huge number of Bag of Words.

**4.** Memory space is the thing where we want to concentrate for multiple documents.

**5.** Multi-document summaries are produced from multiple documents and they have to deal with three major problems:

I. recognizing and coping with redundancy;

II. Identifying important differences among documents;

III. Ensuring summary coherence.

## 2.2 Document Summarization

Due to the limitations in natural language processing technology, abstractive approaches are restricted to specific domains. In contrast, extractive approaches commonly select sentences that contain the most significant concepts in the documents. These approaches tend to be more practical. Recently various effective sentence features have been proposed for extractive summarization, Such as signature word, event and sentence relevance. Although encouraging results have been reported, most of these features are investigated individually. We argue that it is ineffective to identify sentence importance from a single point of view. Each sentence feature has its unique Contribution and combing them would be advantageous. Therefore we investigate combined sentence features for extractive summarization. [2]

Currently, most successful multi-document summarization systems [5] follow the extractive summarization framework. These systems first rank all the sentences in the original document set and then select the most salient sentences to compose summaries for a good coverage of the concepts. For the purpose of creating more concise and fluent summaries, some intensive post-processing approaches are also appended on the extracted sentences.

Two Summary Construction Methods are applied first one is Abstractive method where summaries produce generated text from the important parts of the documents and second is Extractive Method where summaries identify important sections of the text and use them in the summary as they are.

# III. SYSTEM ARCHITECTURE

For implementing graph based document summarization this paper follows following architecture.

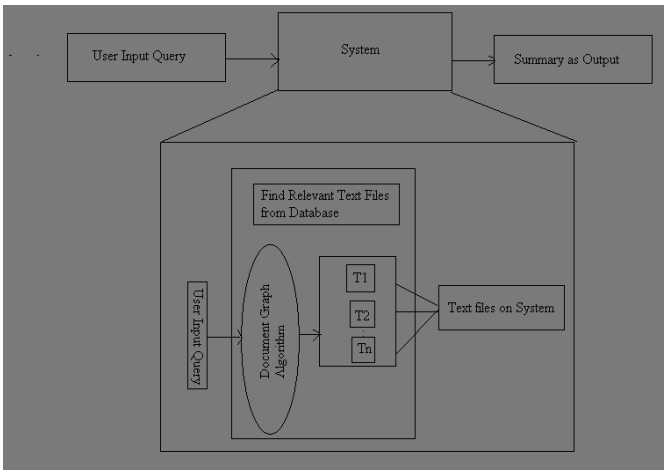


**Fig.1.1 Graph Based System Architecture for Query Specific Summarization**

Document graph is constructed with various phases. Initially all the stop words like (a, an, the, this, that, those, these, is, am, are, were) from all text files will be removed after that document is split into fragments using delimiter paragraph. Every text fragment will be considered as node for the graph. Weighted edge is added between two nodes if they are semantically related. Graph of each document is made. For minimizing the complexity of graph we are considering intermediate nodes with some threshold value. We have taken threshold value as 0.5. With different combination of nodes we are getting various spanning trees. Our work is to consider all combinations of spanning tree and calculate their score. Whichever spanning tree will generate smallest score that will be our summary.

Above fig shows document graph algorithm which is applied on text file i.e. T1, T2, Tn. system will show result as per user input query. To make a document graph and calculating the node weight IR ranking algorithm i.e. okapi [1] is referred which is based on tf-idf principle.

$$\sum_{t \in Q,d} \ln \frac{N - df + 0.5}{df + 0.5} \cdot \frac{(k_1 + 1)tf}{(k_1(1-b) + b\frac{dl}{avdl}) + tf} \cdot \frac{(k_3 + 1)qtf}{k_3 + qtf} \quad \text{……. .(1)}$$

tf: it is a term frequency in document.

qtf is term frequency in user input query.

N is the total number of documents in collection.

df is the number of documents that contains the query term.

dl is document length.

avdl is average document length.

K1 between 1.0-2.0,b is 0.75 and k3 is 0 to 1000. Consider k1, K2, k3 as constants.

Above formula is implemented through programming and nodes from text files are created.

### 3.1 Problem definition

Lets we have n document i.e.d1, d2, to dn. Size of document is total number of words. i.e.size (di).

Term frequency tf(d,w) is no of words present in documents.

Inverse document frequency is i.e.idf(w) Means inverse of documents contain word w in all documents.

Keyword query is set of words. i.e.Q(w1,w2…wn).

The document graph G(V,E) of a document d is defined as follows:

• d is split to a set of non-overlapping text fragments t(v),each corresponding to a node v€V.

• An edge e(u,v) €E is added between nodes u,v€V if there is an association between t(u) and t(v) in d.[1]

### 3.2 System Implementation

For implementation of this system we made various modules by using programming language JAVA. And for storing document graph ORACLE is used. Whole system is divided into various modules.1. Add Remove module 2.Stop Word Elimination Module 3.Document Graph Generation Module 4. Summary Module. This system is executed in the network environment so administrator rights are also considered to monitor the system.

Add Remove module is responsible for adding and removing various text files on the system. Stop word module will remove all the stop word from all the text files as well as from input query which is given to it. Document Graph Generation module will generate the document graph of each text file and that will be stored in ORACLE database. Document graph is generated only once and it is stored in database. If administrator wants to add new files or remove files then he or she want to execute all the modules from starting and make sure that by removing stop word all files nodes are stored inside the database. It means that whenever text files are added or removed from specific drive whole system should be run from the beginning to build a document graph of new files. Summary module will compare input user query with all documents graph and generates the summary.

On text file nodes are created by considering one paragraph as one node likewise all nodes are created for specified file. To make a connection between these nodes i.e. for making edges we are using following formula. [1]

$$EScore(e) = \frac{\sum_{w \in (t(u) \cap t(v))} \left( (tf(t(u), w) + tf(t(v), w)) \cdot idf(w) \right)}{size(t(u)) + size(t(v))} \quad \ldots\ldots\ldots \quad (2)$$

Summary module is referring the concept of spanning tree on the document graph because multiple nodes may have input query so which nodes will be selected? Different combinations from graph are identified and score is generated using following formula. [1]

$$\text{Score}(T) = a \sum_{edge\ e \in T} \frac{1}{EScore\ (e)} + b \frac{1}{\sum_{node\ v \in T} NScore\ (v)} \quad \ldots\ldots\ldots (3)$$

Given the document graph G and a query Q, a summary (sub tree of G) T is assigned a score Score (T) by combining the scores of the nodes v € T and the edges e € T. Where a and b are constants (we use a=1 and b=0.5), EdgeScore(e) is the score of edge e using equation 2, NodeScore(v) is the score of node v using Equation 1.

## IV. PRACTICAL EXAMPLE

Above fig1.1 shows a System for Document Summarization. User will input the query and system will display the summary of text files which are related to user query. We are showing how

summarization process works for text file. Here Fig 1.2 indicates a file which contains n number of paragraphs or sentences.

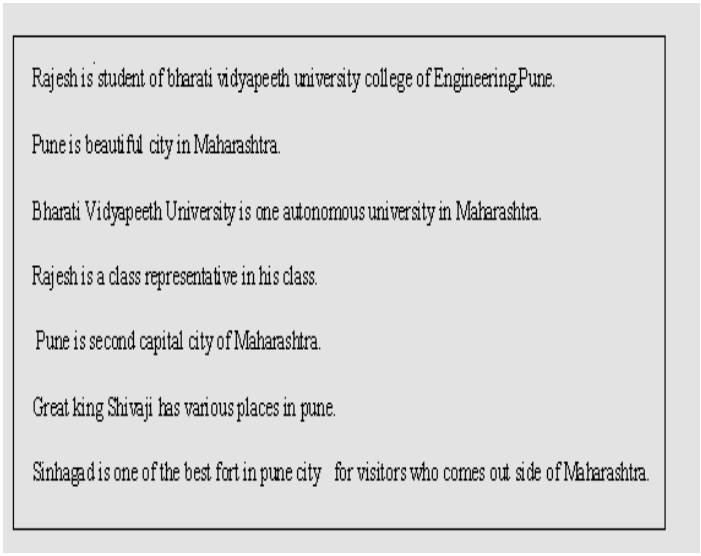


**Fig.1.2 Text File with N Paragraphs**

The Document graph for above file can be created by considering individual line as a single node and then edges can be connected by comparing two lines which have word similarity.Fig.1.3 shows Document Graph for above text file. Information Ranking Techniques are used for giving weights to edges.

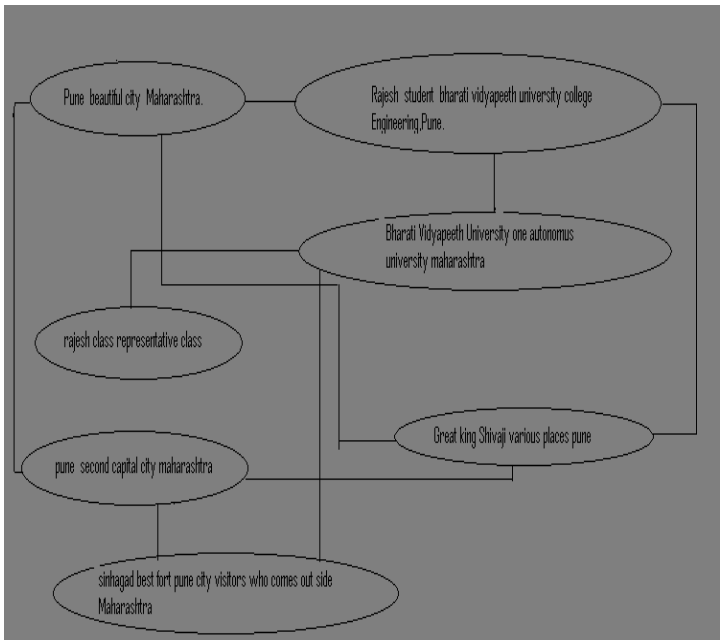


**Fig.1.3 shows Document Graph for above text file**

Suppose Query is **Pune city.**

Then this query is compared with all nodes and accordingly nodes will be selected which contains input query words. From above graph various combinations will be obtained. In text file we have total

seven lines so we are getting only seven nodes. Document graph is made using formula (1) and (2).score of each spanning tree is calculated using formula (3).

**After removing Stop words we get following result.**

Rajesh student bharati vidyapeeth university college Engineering, Pune.

Pune beautiful city Maharashtra.

Bharati Vidyapeeth University one autonomous university Maharashtra.

Rajesh class representative class.

Pune second capital city Maharashtra.

Great king Shivaji various places pune.

Sinhagad best fort pune city visitors who comes out side Maharashtra.

**Out Put**

**In this phase-I the nodes which has input query that sentences will be selected.**

**Phase-I Process**

Rajesh is student of bharati vidyapeeth university college of Engineering,Pune.

Pune_is beautiful city in Maharashtra.

Great king Shivaji has various places in pune.

Sinhagad is one of the best fort in pune city for visitors who comes out side of Maharashtra.

In generating summary in final phase score of each spanning tree is calculated and which spanning tree is having less is displayed at first position likewise all spanning tree will be displayed in asending order with summary of text file.

**Output in final phase.**

Pune is beautiful city in Maharashtra.

Sinhagad is one of the best fort in pune city for visitors who comes out side of Maharashtra.

# V. EXPERIMENTAL RESULT

We have run above program on the system with Hardware configuration as follows.

Pentium Processor: –IV, Hard disk: 160 Gb, RAM Capacity: 1 Gb

Software requirement for implementation of above system is

Operating System: Windows XP, JDK 1.6, ORACLE 9 i

We have stored 57 text files in the database, the memory capacity required for these text files were 122 kb.

After running this program we get various results that are shown in table 1.0.

Following table shows the input query and result for the system. First query is “Network” this keyword is searched with all 57 files. Out of 57 files 11 files contains that keyword so that 11 text

files will be considered as documents for generating summary. Starting from first file system may get different paragraph with keyword "Network" .scores of each combination is calculated likewise all spanning tree combinations scores will be calculated for each file. Among these all document which spanning tree has minimum score that is displayed first. After running this system first summary score is displayed as 11.5044016. And time required for input query is 7 seconds. Score is mainly depending on size of file, number of files which are present in the database. Score are calculated using above equation and spanning tree algorithm. Table 1.0 indicates result for input query "Soft", "Software", "Computer", "System". In this fashion score for input query is calculated. If we use different information ranking formulas and TOP-1 expanding search algorithm definitely the time required for execution can be reduced.

| Query Keywords | No of file contains keyword | Output time In second | Score |
|---|---|---|---|
| Network | 11 | 7 | 11.5044016 |
| Soft | 1 | 6 | 100.744885 |
| Software | 11 | 8 | 3.6043922 |
| Computer | 22 | 9 | 3.9982438 |
| System | 24 | 6 | 3.3116584 |

**Table 1.0 Score of Input query**

# VI. CONCLUSION

Query Summarization using graph based algorithm techniques can be applied over internet, intranet and desktop systems for accessing various document files with short access time. Document Graph based algorithms are initially applied over all text files and user query is applied on this document graph so execution time is also reduced. These techniques can be applied with Google, MSN, Yahoo search engines and user waiting time for accessing document files can be reduced in the future by using top1 ranking algorithm performance of graph based systems can be improved.

## ACKNOWLEDGEMENTS

I am thankful to Professor & H.O.D. Dr. S. H. Patil, Associate Professor M. S. Bewoor, Prof. Shweta Joshi for their continuous guidance. I am also thanks to all my friends who are directly or indirectly supported me to complete this system.

# REFERENCES

[1] Ramakrishna Varadarajan School of Computing and Information Sciences Florida, "International University, paper on "A System for Query-Specific Document Summarization**".**

[2] Pinaki Bhaskar and Sivaji Bandyopadhyay Department of Computer Science & Engineering, Jadavpur University, Kolkata – 700032, India, "A Query Focused Multi Document Automatic Summarization".

[3] Kam-Fai Wong*, Mingli Wu,Department of Systems Engineering and Engineering Management The Chinese University of Hong Kong, "Extractive Summarization Using Supervised and Semi-supervised Learning".

[4] You Ouyang, Wenji Li, Qin Lu Department of Computing, the Hong Kong Polytechnic University, "An Integrated Multi-document Summarization Approach based on Word Hierarchical Representation".

[5] Rakesh Agrawal, King-Ip Lin, Harpreet S. Sawhney, and Kyuseok Shim, "Fast similarity search in the presence of noise, scaling, and translation in time-series databases". In Proc. of the VLDB Conference, Zurich, Switzerland, September 1995.

[6] Neill Alexander, Craig Brown, Joemon Jose, Ian Ruthven1 and Anastasios Tombros Department of Computing Science University of Glasgow, Glasgow, G12 8QQ. Scotland, "Question answering, relevance feedback and summarization".

[7] Xiaojun Wan and Jianguo Xiao, Institute of Computer Science and Technology Peking University, Beijing 100871, China, "Graph-Based Multi-Modality Learning for Topic-Focused Multi-Document Summarization".

[8] Yajie Miao, Chunping Li School of Software Tsinghua University, Beijing 100084, China,Enhancing "Query-oriented Summarization based on Sentence Wikification".

[9] Balabhaskar Balasundaram 4th IEEE Conference on Automation Science and Engineering Key Bridge Marriott, Washington DC, USA August 23-26, 2008 "Cohesive Subgroup Model For Graph-based Text Mining"

**Authors**

**Prashant D. Joshi** is a student of M.Tech Sem IV Student in Computer Engineering, Bharati Vidyapeeth Deemed University College of Engg, Pune-43. He is also working as a Assistant Professor in Department of Computer Engineering and having total 6 years of teaching experience.

**M. S. Bewoor** working as an Associate Professor in Computer Engineering Bharati Vidyapeeth Deemed University college of Engg, Pune-43.She is having total 10 years of teaching experience.

**S. H. Patil** working as a Professor and Head of Department in Computer engineering, Bharati Vidyapeeth Deemed University college of Engg,Pune-43. He is having total 22 years of teaching experience & working as HOD from last ten years.