Document Summarization Techniques

Simran kaur[1], wg.cdr anil chopra[2]

[1]assistant professor, lingayas university Faridabad
[1]simrankaurjolly@gmail.com

[2] Wg.Cdr AnilChopra,

Manav Rachna International University Fbd
[2]wgcdranilchopra.fet@mriu.edu.in

Abstract— Text summarization as we know today is widely used everywhere to condense our data in a single summary. Text summarization is basically of two types: a) extractive summarization-where important text segments are taken from original text. b) Abstractive summarization: where original text is interpreted and presented you in consolidated form. Now a day's research is going on in the area of multi-document summarization where clusters of documents are formed and are summarized on the basis of their distance from central document (centroid).the meaning of the words can also be taken from largest online dictionary word net and their tf*idf values can be computed easily which is essential part of sentence pre-processing.
Keywords- Document Summarization, Sentence Pre-processing, Natural Language processing (nlp), Clustering, Word Net.

Introduction

The World Wide Web has brought us a vast amount of on-line information. Due to this fact, every time someone searches something on the Internet, the response obtained is lots of different Web pages with huge information, which is impossible for a person to read completely. As the information resources in both online and offline are increasing exponentially, the major challenge is to find relevant information from large amount of data. Text summarization is an effective technique that is used in combination with Information Retrieval and Information filtering systems to save the user time. This is the same case with the blogs. Users do not want to read the whole blog post. Rather want the brief summary. Summary of the document can be helpful to the user to get the main theme of the document in a short span of time. Text summarization is the process of compressing the original document into a short summary by extracting the most important information from the document. Likewise blog summarization is a way of compressing the blog post to get an extract from the blog post.

Much existing research on blogs focused on posts only, ignoring their comments. But readers treat comments associated with a post as an inherent part of the blog post. Reading comments does change one's understanding about blog posts. We aim to extract representative sentences from a blog post that best represent the topics discussed among its comments.

Text summarization [2] is the process of extracting important information from a given text. Based on the how this important information is presented to the user, two types of text summarization systems are defined. They are 1) Extractive summarization system 2) abstractive summarization system. In Extractive in Extractive Summarization system important text segments of the original text are identified and presented as they are. In abstractive summarization original text is interpreted and is written in a condensed form so that the resulting summary contains the essence of the original text. The summary in extractive summarization contains the words and sentences of the original text. This may not happen in abstractive summarization system. Stop lists play an important role in building search engines and text summarization systems. They help in filtering useful information from the original text. Traditional Stop lists are those which are specific to a natural language and are primarily developed for use in a search engine. Since Text summarization is a complex task involving natural language processing, it uses natural language processing tools like Dictionaries, Thesaurus, Word net, and POS Tagger etc... .A Parts-Of-Speech Tagger (POS Tagger) is a piece of software that reads text in some language and assigns parts of speech to each word (and other token), such as noun, verb, adjective, etc., although generally computational applications use more fine-grained POS tags like 'noun-plural'. Word net is an on-line lexical reference system whose design is inspired by current psycholinguistic theories of human lexical memory. Second one is a Stanford Log-linear Part-Of-Speech Tagger. Term frequency is a statistical measure used in calculating relevance of a document. It tells something about the document as a whole with respect to a user query. Many document summarization methods are based on conventional term weighting approach for picking a set of frequencies and term weights based on the number of occurrences of the words is calculated. Summarization methods based on semantic analysis also use term weights for final sentence selection. The term weights generally used are not directly derived based on any mathematical model of term distribution or relevancy. In our approach, we use a term frequency model to mathematically characterize the relevance of terms in a document. This model is then used to extract important sentences from the documents. Another major issue to be handled in our study is to generate a "user-friendly" summary at the end.

I. SENTENCE PRE-PROCESSING ON BLOGS

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, and that is no longer than half of the original text. Text summarization is the process of distilling the most important information from a source to produce an abridged version for a particular user and task.

Blog summarization is the process of reducing the blog post in order to create a summary that retains the most important points of the original blog.

With the rapid growth of World Wide Web, people tend to write blogs to share their feelings and experiences. However, with overloaded information, readers do not have time to go through every detail. And yet they want to know other people's opinion towards certain topic. In this project, we build a blog summarization system to assist people quickly getting opinion from vast amount of online blog information. A system is built up based on the comment oriented summarization. In contrast to existing summarization systems, which make factual summaries, our system takes into consideration the characteristic of blog. After identifies sentences that relevant to reader's preference by considering the comment for summarizing.

A. Selecting a Template (Heading 2)

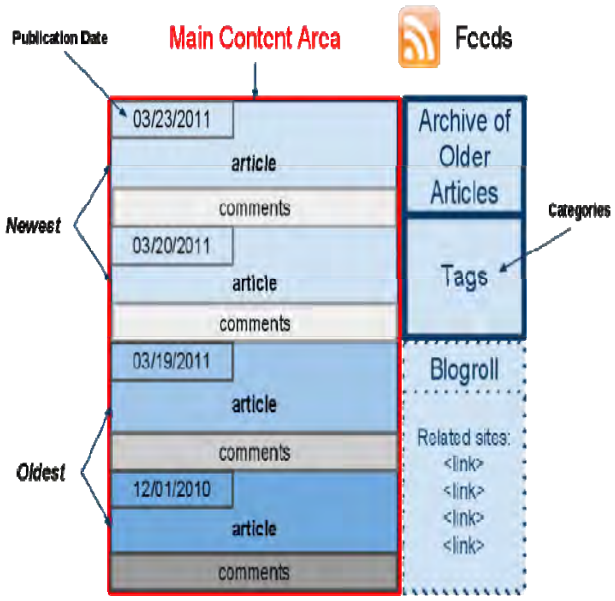

Figure 1: a structure of a blog

A. Existing System

Existing systems include text summarizations. Text summarization has traditionally been focused on text input. Various methods for summarization were proposed which include extraction-based, abstraction-based, maximum entropy-based and aided summarization. These methods use linguistic and natural language processing techniques. The steps followed are interpretation of the source text to obtain a text representation, transformation of the text representation into a summary representation, and finally, generation of the summary text from the summary representation.

Most of the existing methods for document summarization use the information present in the given document. These methods have explored various techniques for summarization process based on the assumption that the specified document is independent of any other documents. But the few topic related documents can be helpful for producing summary. The alternate approach for document summarization will make use of neighbor documents which provide the neighborhood knowledge that makes the summarization process efficient.

Comments left by readers on Web documents contain valuable information that can be utilized in various information retrieval tasks including document search, visualization and summarization. In this project, we study the problem of comments-oriented blog summarization and aim to summarize a blog post by considering not only its content, but also the comments left by its readers. Much existing research on blog summarization focused on posts only, ignoring their comments. Reading comments does change one's understanding about blog posts. In this project, we aim to extract representative sentences from a blog post that best represent the

topics discussed among its comments. The proposed solution first derives representative words from comments and then selects sentences containing representative words. Significant differences between the sentences labelled before and after reading comments can be observed.

By considering these comments, the generated summary can better capture the input from the readers, as opposed to the author of the blog only. That is, a comments-oriented summary provides balanced views from both author and readers. Second, most websites present a blog post together with its comments. Also readers treat comments associated with a post as an inherent part of the post. A comment -oriented summary hence better matches one's understanding of the blog post as readers often read the post together with its comments. Third, the generated summary could better support many IR applications.

The system works like:

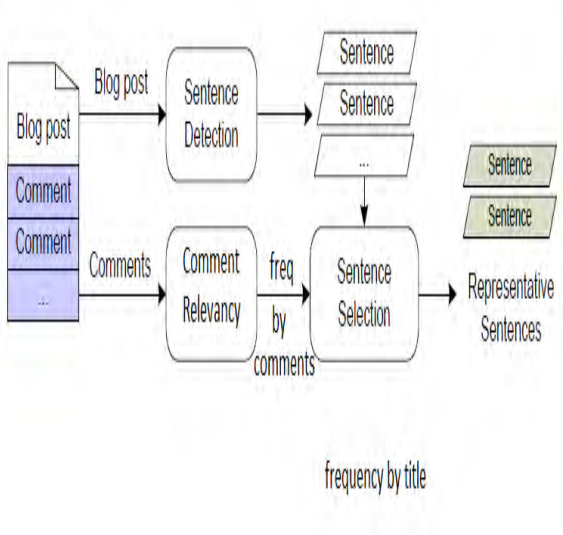

Figure 2

B. Functional Components

Following is a list of the functional components of the tool.

1) Text pre-processor: This will work on the HTML documents and convert them to plain text for processing by the rest of the system.

2) Title Extractor: This extracts the title of the blog whose words are used as the keywords to evaluate the frequency of each sentence.

3) Sentence separator: This goes through the blog post and separates the sentences based on some rules (like a sentence ending is determined by a dot and a space etc). Any other appropriate criteria might also be added to separate the sentences.

4) Word separator: This separates the words based on some criteria (like a space denotes the end of a word etc).

5) Stop-words eliminator: This eliminates the regular English words like 'a, an, the, of, from' etc for further processing. These words are known as 'stop-words'. A list of applicable stop-words for English is available on the Internet.

6) Duplicate remover: This ensures the distinctness in the list of keywords. It removes all duplicates and retains single copy of the words.

7) Stemmer: Various stemming rules can be applied to convert the words back to their root form. For example 'eats' can be converted to 'eat' using stemming. Some of the rules used are:

- SSES → SS
- IES → I
- SS → SS
- S → ' '

8) Word-frequency calculator: This calculates the number of times a word appears in the document (stop-words have been eliminated earlier itself and will not figure in this calculation). For example, the word 'Unix' may appear a total of 100 times in a document, and in 80 sentences. (Some sentences might have more than one occurrence of the word). This calculator calculates the times (frequency) of occurrence of various keywords in each sentence. Keywords may include title words or comment words.

9) Scoring algorithm: This algorithm determines the score of each sentence. Several possibilities exist. The score can be made to be proportional to the sum of frequencies of the different words comprising the sentence (i.e. If a sentence has 3words A, B and C, then the score is proportional the sum of how many times A, B and C have occurred in the document). The score can also be made to be inversely proportional to the number of sentences in which the words in the sentence appear in the document. Likewise, many such heuristic rules can be applied to score the sentences. In this project, we are using a 70-30 ratio of the keywords frequency, 70% to the frequency of the title keywords and 30% for the keywords extracted from the comment.

10) Ranking: The sentences will be ranked according to the score of their frequency.

11) Summarizing: Based on the user input of the size of the summary required, the sentences will be picked from the ranked list and concatenated. The resulting summary will be displayed on screen.

II. PROCEDURE

Steps followed in the process of summarization are:

1. File containing the source code is analyzed and title of the blog is extracted.
2. Stop words and duplicates are removed from the title. Also stemming rules are applied on the extracted title
3. The body of the blog post is now extracted to get the content of the blog post.
4. Stemming of the body content is done and stop words are removed.
5. All the sentences in body are separated using the sentence separator.
6. The frequency of each word present the title is evaluated for each sentence using the frequency evaluator. Finally the total frequency (f1) of all keywords of title is evaluated for each sentence present in body.
7. Comments of the blog post are now extracted.
8. Each comment is now checked for its relevancy. If the comment has some similarity with the title keywords it is considered relevant else irrelevant.
9. Only relevant comments are considered for further processing and rest are discarded.
10. Relevant comments are stemmed; duplicates and stop words are removed.
11. uency of each sentence of body is evaluated corresponding to every word present in comment. Finally the total frequency (f2) of all keywords of comment is evaluated for each sentence present in body.
12. Overall frequency(F) for each sentence is calculated using F = (0.7*f1) + (0.3*f2) This frequency measure is used to rank the sentences.
13. The number of sentences for summary is input by the user.
14. The top ranked sentences are output to the user.

III. RELATED WORK ON CLUSTERING

Clustering is an important technique used in areas such as information retrieval, text mining, and data mining. Clustering algorithms combine data points into groups such that: (i) data points in the same group are similar to each other; and (ii) data points in one group are “different” from data points in a different group or cluster. In information retrieval it is assumed that documents that are similar to each other are likely to be relevant for the same query, and therefore having the document collection organized in clusters can provide improved document access. Different clustering techniques exists the simplest one being the one-pass.

A. Clustering Algorithm

We have implemented an agglomerative clustering algorithm which is relatively simple, has reasonable complexity, and as it will be shown gave us good results. Our algorithm operates in an exclusive way, meaning that a document belongs to one and only one cluster— while this is our working hypothesis, it might not be valid in some cases. The input to the algorithm is a set of document representations implemented as vectors of terms and weights (term_1 D weigth_1; term_n D weigth_n). Initially, there are as many clusters as input documents; as the algorithm proceeds clusters are merged until a certain termination condition is reached. The algorithm computes the similarity between vector representations in order to decide whether or not to merge two clusters.

The similarity metric [3] we use is the cosine of the angle between two vectors. This metric gives value one for identical vectors and zero for vectors which are orthogonal (no related). Various options have been implemented in order to measure how close two clusters are, but for the experiments reported here we have used the following approach: the similarity between two clusters (simC) is equivalent to the similarity (simD) between the two more similar documents in the two clusters—this is known as single linkage in the clustering literature; we take simD to be the cosine metric computed as follows:

Cosine (d1, d2) =w1d1*w2d2/ $(w1d1)^2$sqrt*$(w2d2)^2$sqrt

Here d1 and d2 are document vectors and wi; dk is the weight of term i in document dk. If this similarity

between two clusters is greater than a threshold—experimentally obtained—they are merged together. Each iteration in the algorithm the most similar pair of clusters is merged. If this similarity is less than a certain threshold the algorithm stops. Merging two clusters consist of a simple step of set union, so there is no re-computation involved—such as computing a cluster centroid.

B. Proposed Methodology

So we presented our approach towards 'k means clustering Automated Text Summarization'. Our approach attempts to generate a text summary from the article of newspapers, while avoiding the repetition of identical or similar information and presenting the information in such a way that makes sense to the reader. The proposed algorithm work as follows in fig6:

1. S= (D1...Dn) here's is a collection of all documents which are related to same topic.
2. We use vector of tokens to represent each documents T= (t1...tn).
3. These tokens are the words appearing frequently in document except some stop words and he threshold value is determined for each document for the words to be taken.
4. After getting tokens or words from each documents we get tf*idf values or weights of the words using word net dictionary.
5. After we get weighted measures of all the words which is denoted by V(d1)=(wi1…win) where win=TF(di,tj).log(N/DF(tj))
6. Here TF stands for term frequency which tells how many times a term occurs in documents divided by total number of terms in documents.
7. IDF stands for inverse document frequency which is log_e (total no of docs/no of documents with term t in it).
8. After getting tf*idf values of all the tokens we group them in cluster by k means clustering algorithm. Here we initially take three tokens randomly as initial centroids.
9. For k=no of desired clusters i.e. 3 we take three values r1,r2,r3 where each of them have a particular tf.idf value
10. Then we compare the Euclidean distance r1f1=d11, r2f1=d21, r3f1=d31.
11. If d11<d21 && d11<d31 then f belongs to first cluster i.e. c1 or
12. D21<d11 && d21<d31 then f1 belongs to c2
13. These iterations continue till the epoch value which is no of iterations given i.e. or mean value of all the d1...dn values is nearly equal to mean of all k means which is the centroid for all documents.
14. After we get a stable clusters for all the tokens then they are grouped according to their frequencies in the respective clusters c1,c2,c3
15. After we get clusters and all tokens arranged in clusters the sentences having the words are selected in clusters from documents d1…dn.
16. After applying summarizer on all the three clustered documents we get summaries of the cluster on the basis of word scoring.

K-means algorithm

1. Input:

D = d1, d2..... dn // set of n data items or tokens k // Number of desired clusters

2. Output:

A set of k initial centroids.

3. Steps:

a. .Calculate the average score of each data point;

b. di=x1,x2,x3,..,xn

c. di(avg)=(w1*x1+w2*x2+w3*x3+ ..... +wm*xm)/m // where x = the attributes value, m = number of attributes and w = weight to multiply to ensure fair distribution of cluster Sort the data based on Euclidean distance from the k centroids;

a. Calculate Euclidean distance of the data points from the centroid which is $d^2(xn,xk)$

b. Divide the data into clusters;

c. Calculate the mean value of the each cluster;

d. Take the nearest possible data point of the mean as the initial centroid for each data subsets.

e. The clustering continues until it converges to its stable clusters

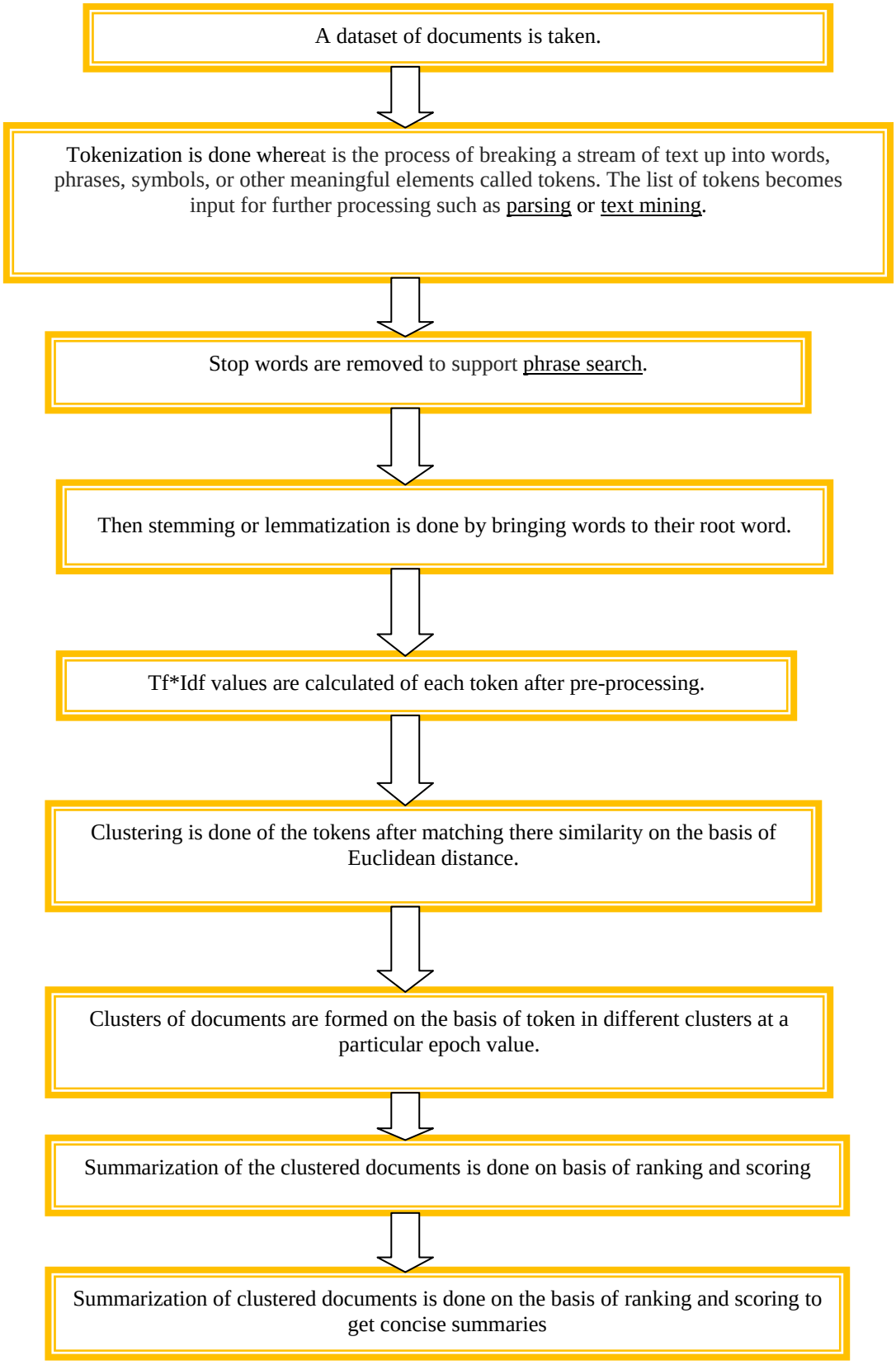

Figure 3: summarization flowchart

Equations

The similarity metric we use is the cosine of the angle between two vectors. This metric gives value one for identical vectors and zero for vectors which are orthogonal (no related). Various options have been implemented in order to measure how close two clusters are, but for the experiments reported here we have used the following approach: the similarity between two clusters (simC) is equivalent to the similarity (simD) between the two more similar documents in the two clusters—this is known as single linkage in the clustering literature; we take simD to be the cosine metric computed as follows:

Cosine (d1, d2) =w1d1*w2d2/ (w1d1)^{2}sqrt*(w2d2)^{2}sqrt

Here d1 and d2 are document vectors and wi; dk is the weight of term i in document dk. If this similarity between two clusters is greater than a threshold—experimentally obtained—they are merged together. Each iteration in the algorithm the most similar pair of clusters is merged. If this similarity is less than a certain threshold the algorithm stops. Merging two clusters consist of a simple step of set union, so there is no re-computation involved—such as computing a cluster centroid.

REFERENCES

[1] Structured and Unstructured Document Summarization: Design of a Commercial Summarizer using Lexical Chains:Hassan Alam, Aman Kumar, Mikako Nakamura, Fuad Rahman1, Yuliya Tarnikova and Che Wilcox BCL Technologies Inc. fuad@bcltechnologies.com.

[2] An intelligent algorithm for document summarization :Yi Guo and George Stylios RiFlex, Heriot-Watt University Scotland, TDI 3HF, U.K. Tel: (+44)1896-892268 {y.guo, g.stylios}@hw.ac.uk.

[3] A Novel Approach to Multi-document Summarization:Li-Qing Qiu Bin Pang Sai-Qun Lin Peng Chen State Key Lab. of Software Development Environment, Beihang University, 100083 {qiuliqing, pangbin, linsq, chenpeng}@nlsde.buaa.edu.cn.

[4] A Context Based Text Summarization System:Rafael Ferreira∗†, Frederico Freitas∗, Luciano de Souza Cabral∗, Rafael Dueire Lins∗, Rinaldo Lima∗, Gabriel Franc¸a∗, Steven J. Simske‡, and Luciano Favaro§ ∗Informatics Center, Federal University of Pernambuco, Recife, Pernambuco,

[5] Comparison of Document Summarization Techniques

[6] A sentence scoring method for extractive text summarization based on Natural language queries :R.V.V Murali Krishna1 and Ch. Satyananda Reddy2.

[7] Extractive Multi Document Summarizer Alogorithim

[8] Quantifying The Limits And Success Of Extractive Summarization Systems Across Domains

[9] Document Summarization as Applied in Information Retrieval.

[10] .A Novel Automatic Text Summarization Study Based On Term Co-Occurrence

[11] Word Net (A Lexical Database for the English Language). Available at http://www.cogsci.princeton.edu/-wn/.

[12] Improvement of K-means Clustering algorithm with better initial centroids based on weighted average

[13] Collaborative Clustering of XML Documents Sergio Greco, Francesco Gullo, Giovanni Ponti, Andrea Tagarelli.

[14] Multi-Document Summarization as Applied in Information Retrieval

[15] A Novel Approach for Organizing Web Search Results using Ranking and Clustering: Neelam Duhan Department of Computer Engineering YMCA University of Sc. & Technology Faridabad, India.

[16] Centroid Integer Selection Model – A High Efficiency Method on Dynamic Multi- Document Summarization

[17] A New Centroid-Based Classifier for Text Categorization

[18] Extracting main content of a topic on online social network by multi-document summarization

[19] Query-focused Multi-document Summarization Using Keyword Extraction Liang Ma1, 2 Tingting He1, 2 Fang Li1, 2Zhuomin Gui1, 2 Jinguang Chen1, 2 (1Department of Computer Science, Huazhong Normal University, Wuhan, China, 430079 2Network Media Branch, National Language Resources Monitoring and Research Center, Wuhan, China, 430079)