Multi Document Summarization
(For News Articles)

A Bachelors Mini Project (6th Semester)

Of the Undergraduate Program in

INFORMATION TECHNOLOGY

Submitted By:

1. Atul Kumar Verma (IIT2012036)
2. Gaurav Kumar Chandel (IIT2012056)
3. Harivansh Kumar (IIT2012026)
4. Mohit Purbey (IIT2012055)
5. Nithin Srikar Karnala (IIT2012014)
6. Ritesh Kumar Sinha (IIT2012003)

Under the Guidance Of:

Prof. Sudip Sanyal, IIIT Allahabad.

Indian Institute of Information Technology Allahabad

Devghat, Jhalwa, Allahabad-211012, U. P. INDIA

Candidate’s Declaration

We hereby declare that the work presented in this project entitled “Multi Document Summarization (For News Articles)”, submitted in partial fulfilment of the Sixth semester of Bachelor of Technology (B.Tech) program, in Information Technology at Indian Institute Of Information Technology, Allahabad, is an authentic record of our original work carried out under the guidance of Prof. Sudip Sanyal and due acknowledgements have been made in the text of the project to all other material used. This work was done in full compliance with the requirements and constraints of the prescribed curriculum.

Place: Allahabad

Date:

Atul Kumar Verma (IIT2012036)

Gaurav Kumar Chandel (IIT2012056)

Harivansh Kumar (IIT2012026)

Mohit Purbey (IIT2012055)

Nithin Srikar Karnala (IIT2012014)

Ritesh Kumar Sinha (IIT2012003)

Acknowledgement

The authors acknowledge the valuable contributions of **Prof. Sudip Sanyal** who guided through this project. His keen awe-inspiring personality, superb guidance and constant encouragement are the motive forces behind this project work. We would like to thank him sincerely for his help and support.

Atul Kumar Verma (IIT2012036)

Gaurav Kumar Chandel (IIT2012056)

Harivansh Kumar (IIT2012026)

Mohit Purbey (IIT2012055)

Nithin Srikar Karnala (IIT2012014)

Ritesh Kumar Sinha (IIT2012003)

Abstract

We present an extractive multi document summarizer that is going to summarize all the news articles related to a single topic from different sources extracted by a web crawler.

We have employed two techniques – centroid based method and graph based method. We have compared the results coming after applying the above methods individually as well as a combination of these methods. The optimum combination has been used for summarization.

Key Terms: Multi document summarization, Centroid based method, Graph based method, NLP.

Table of Contents

Introduction

Now-a-days, we have a lot of information available online. Even on a single topic we can find a large number of news articles from different sources. But in practical scenarios along with the information we are interested in we get lots of noise such as advertisements, related articles and links to other pages. Single document summarization would help but are likely to be similar to each other. So there is a need of an effective mechanism that can provide us the maximum information in minimum amount of time. This can be done using multi document summarization. Ideally, multi document summaries should contain all the key shared relevant information among all the documents only once, plus the other information unique to some of the individual documents that are directly relevant to the user's query.

Multi document summarization techniques are broadly categorized in two types – Extractive and Abstractive. Purely extractive summarizers often give better results than abstractive summarizer as the latter has problems like semantic representation and natural language generation. So we have used extractive summarizer for our project.

Our project is divided into two phases –

Phase 1 – Document collection

> We have employed a web search engine that provides us with all the links related to a particular news topic that the user is interested in.
>
> The new articles are extracted from these links and passed on to a noise removal tool to remove unnecessary information, after this step documents are passed on to phase 2 for summarization.

Phase 2 – Summary Generation

> We have used two methods i.e. Centroid based method and graph based method to generate the summaries. The optimum combination of above two methods has been used for summary generation.

Literature Review

To get the links to desired pages by just providing few keywords, has been evolved up to a saturation level. There are many search engines available that can do the task in milliseconds or less. E.g. Google, Bing. The searching process of web is a tedious task, as it requires ample resources and computation power.

In the field of filtering relevant article out of noisy webpage many approaches have been proposed. For e.g.: Machine learning, Rule based scrapping etc. Some open source API's implementing above concepts are readily available.

Previously most of the work in the field of summarization was done for single document summarization that attempted to deal with the issues by focusing more on a related, but simpler, problem. With text-span deletion the system attempts to delete "less important" spans of text from the original document; the text that remains is deemed a summary. Work on automated document summarization by text span extraction dates back at least to work at IBM in the fifties [1]. Most of the work in sentence extraction applied statistical techniques (frequency analysis, variance analysis, etc.) to linguistic units such as tokens, names, anaphora, etc. More recently, other approaches have investigated the utility of discourse structure [2], the combination of information extraction and language generation [3], and using machine learning to find patterns in text [4][5].

Some of these approaches to single document summarization have been extended to deal with multi-document summarization [6].

Multi document summarizers are of two types –

1. Extractive Summarization:
   Extractive summarizers produce summaries by selecting a subset of sentences from all the documents. There are a number of methods that can be used to select this subset.

2. Abstractive Summarization:
   Abstractive summaries produce summaries by rephrasing the sentences. Abstractive summarizers make extensive use of natural language processing for this task.

Although summaries produced by humans are typically not extractive, most of the summarization research today is on extractive summarization. Purely extractive summaries often give better results compared to automatic abstractive summaries. This is due to the fact that the problems in abstractive summarization, such as semantic representation, inference and natural language generation, are relatively harder compared to a data-driven approach such as sentence extraction. In fact, truly abstractive summarization has not reached to a mature stage today.

Early research on extractive summarization is based on simple heuristic features of the sentences such as their position in the text, the overall frequency of the words they contain, or some key phrases indicating the importance of the sentences [7]. A commonly used measure to assess the importance of the words in a sentence is the *inverse document frequency*, or *idf*, which is defined by the formula [8]:

$$idf \ = \log(N/n_i)$$

Where *N* is the total number of the documents in a collection, and n_i is the number of documents in which word *i* occurs. For example, the words that are likely to occur in almost every document (e.g. articles “a” and “the”) have *idf* values close to zero while rare words (e.g. surgical terms, proper nouns) typically have higher *idf* values.

More advanced techniques also consider the relation between sentences or the discourse structure by using synonyms of the words or anaphora resolution [9]. Researchers have also tried to integrate machine learning into summarization as more features have been proposed and more training data have become available [10].

Our summarization approach in this paper is to assess the centrality of each sentence in a cluster and extract the most important ones to include in the summary.

Methodologies

Phase 1 – Document Collection

The task of gathering relevant news articles to give as an input to our multi document summarizer has two parts -

1. Finding the webpages containing the news articles over web, given the search terms for the news topic.
2. Extracting the relevant article leaving out the noisy data such has headers, comments and other irrelevant sections.

Working of search Engines –

The process of searching has mainly three parts.

- Crawler.
- Indexing.
- Query Processing.

Crawler - is automated program which runs continuously in background doing following task.

1. Pop out an address from inbuilt queue and visit the page.
2. Extract all text and links out of the page.
3. Send the text to Indexer and push all links back in to the queue.
4. Go to step a. and repeat the process until some threshold is reached.

Various open-source crawlers available are Crawler4j, JSOUP, Scrapy, Selenium.

Indexing - The Process of Indexing consists of following steps.

1. Converting all the words to lower case letters.
2. Removing the stop-words (e.g.: in, the, by, of etc.)
3. Stem the words (Porter stemmer may be used.)
4. Now calculate the frequencies of each word.
5. Maintain a pair of word and list of documents sorted in the order of count of occurrence of that word.
6. Each of the document id is also paired with, the frequency of that word. E.g.: <India, <doc3, 6>, <doc1, 4>, <doc7, 2>>

Query Processing - Query processor involves following steps.

1. Tokenize the query terms.
2. Remove stop words.
3. Stem them using any stemmer.
4. Go to indexed data structure and return the intersection doc-ids which have maximum time occurrence of the given words.
5. Return the corresponding URL’s to user.

Methods that we are going to use are as follows -

1. **Searching**: As our main focus is on the problem of multi-document summarization, we did not want to develop our own search engine, as the problem of searching itself requires great amount of computational resources.

   For the task of searching we are using the open source API of Bing, which can be embedded inside the program of Multi-document Summarization.

   Bing API gives features to the developers to build application that enables to: Retrieve Valuable Information from the Search engine when needed, get instant answers to questions, retrieve telephone numbers for the users, monetize applications with advertisements and Add Instant News Articles and much more.

2. **Extracting**: For the extraction of relevant part, we are using an open source API, Boilerpipe. Boilerpipe provides algorithms to detect and remove the surplus "clutter" (boilerplate, templates) around the main textual content of a web page based on shallow text features.

Phase 2 - Multi document summary Generation

Extractive summarization relies on the concept of sentence salience to identify the most important sentences in set of documents that will be included in the final summary. Salience is typically defined in terms of the presence of particular important words or in terms of similarity to a centroid pseudo sentence.

Methods we are going to employ to find salient sentences in a set of documents are –

1. **Centroid Based**

   Extractive summarization process can be viewed as identifying the most *central* sentences in a (multi-document) cluster that give the necessary and sufficient amount of information related to the main theme of the cluster.

   Centrality of a sentence is often defined in terms of the centrality of the words that it contains. A common way of assessing word centrality is to look at the centroid of the document cluster. The centroid of a cluster is a pseudo-document which consists of words that have **tf** × **idf** scores above a predefined threshold, where *tf* is the frequency of a word in the cluster, and *idf* (inverse document frequency) values are typically computed over a much larger and similar genre data set. In centroid-based summarization [11], the sentences that contain more words from the centroid of the cluster are considered as central. This is a measure of how close the sentence is to the centroid of the cluster. The centrality of a sentence can be formulated as the number of words occurring both in the sentence and centroid of the cluster.

2. **Graph Based** (Using cosine similarity between any two sentences) –

   The cluster of documents extracted after phase 1 can be considered as a network of sentences that are related to each other. Some sentences will be more similar to each other while some others may share only a little information with the rest of the sentences. We assumed that sentences that are similar to many other sentences in the obtained cluster are more central (salient) to the topic.

   Defining similarity between two sentences:

To define similarity, we use the bag-of-words model to represent each sentence as an N -dimensional vector, where N is the number of all distinct words in the obtained cluster. For each word that occurs in a sentence, the value of the corresponding dimension in the vector representation of the sentence = ***number of occurrences of the word in the sentence x idf*** of the word.

Now the similarity between any two sentences is given by the cosine between the vectors of the corresponding two sentences:

$$\text{Cosine(x, y)} = \frac{\sum_{w \in x,y} f_{w,x} f_{w,y} (idf_w)^{\,2}}{\sqrt{\sum_{x_i \in x} (f_{x_i,x} idf_{x_i})^2} \times \sqrt{\sum_{y_i \in y} (f_{y_i,y} idf_{y_i})^2}}$$

Where $f_{w,s}$ is the number of occurrences of word w in sentence s.

The obtained cluster of documents from phase 1 may now be represented by a cosine similarity matrix where each entry in it is the similarity between the corresponding sentence pair.

Now to compute the overall centrality of a sentence given its similarity to other sentences:

In the obtained cluster, many of the sentences will be similar to each other since they are all about the same topic. Therefore the majority of the values in the similarity matrix will nonzero. Since we are interested in *relevant* similarities, we can eliminate some low values in this matrix by defining a threshold so that the cluster can be viewed as an (undirected) graph, where each sentence of the obtained cluster is a node, and significantly similar sentences are connected to each other.

Now the centrality of a sentence can be defined as the degree of the node of the corresponding sentence, the higher degree implies that the more it is related to other sentences there by the more important or significant it is.

We have combined the above two approaches and calculated the combined centrality of the sentences. The centrality of the sentence is then formulated as the weighted sum of the above two approaches. Let the weights of centroid based and graph based be α and β respectively. Then the combined hybrid centrality of a sentence is nothing but α times centrality of the sentence based on centroid approach plus β times the centrality of the sentence based on degree (graph) approach.

Now after calculating the hybrid centrality of each sentence we picked up the sentences based on their hybrid centrality. Number of sentences to be selected have been decided based on given word limit or sentence limit or any such constraint.

We will be using data sets of DUC (Document Understanding Conferences) in our experiments since Task 2 of DUC involve generic summarization of news documents clusters. All data sets are in English.

After applying above two methods we get a bag of sentences with a hybrid rank assigned to each sentence. Now to order the sentences in chronological order, we calculated the normalized position value of each sentence and combined this value with the hybrid ranking obtained previously. This gives a ranking which is based on importance and position of the sentence in a document.

After all this processing we output the summary of desired size based on this ranking.

Results

To obtain the most appropriate values of thresholds and multiplying factors while calculating rankings, we played around the values to see the changes in the summary generated. In this section we are presenting our results.

a) TF x IDF values

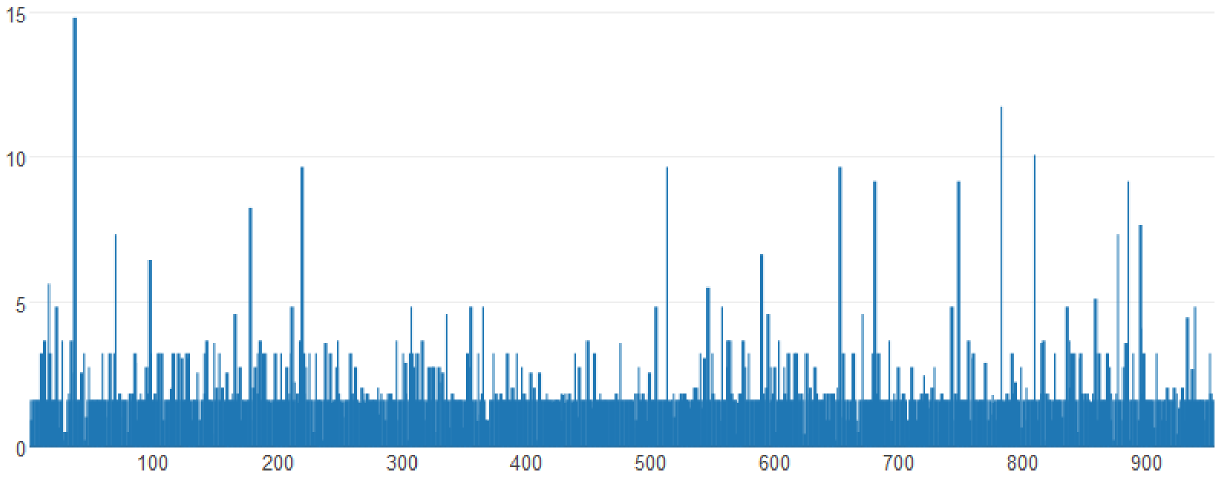

Fig. 1 – Bar plot of tf x idf values for Net Neutrality data

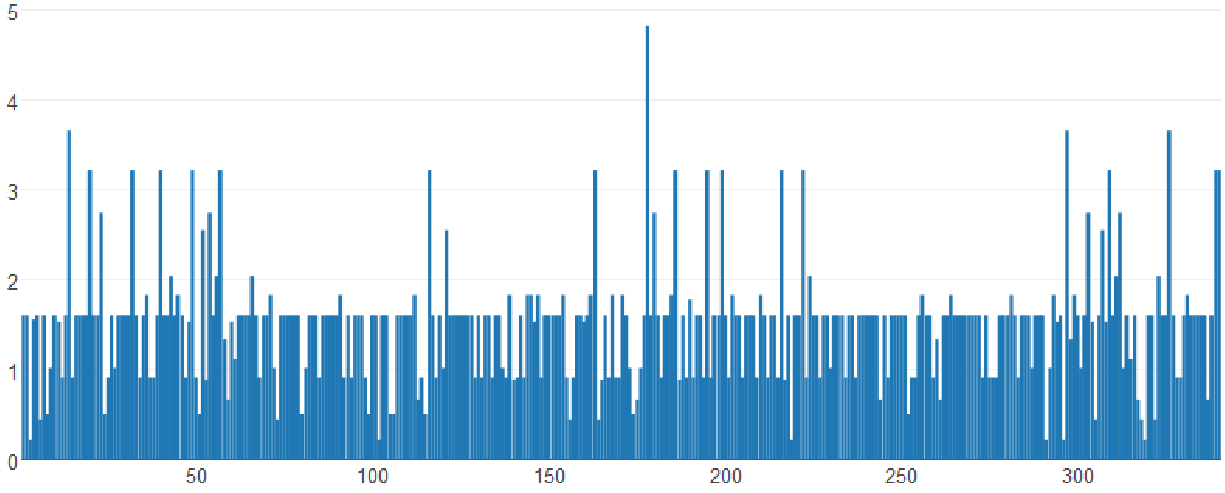

Fig. 2 – Bar plot of tf x idf values for Farmer Suicide data

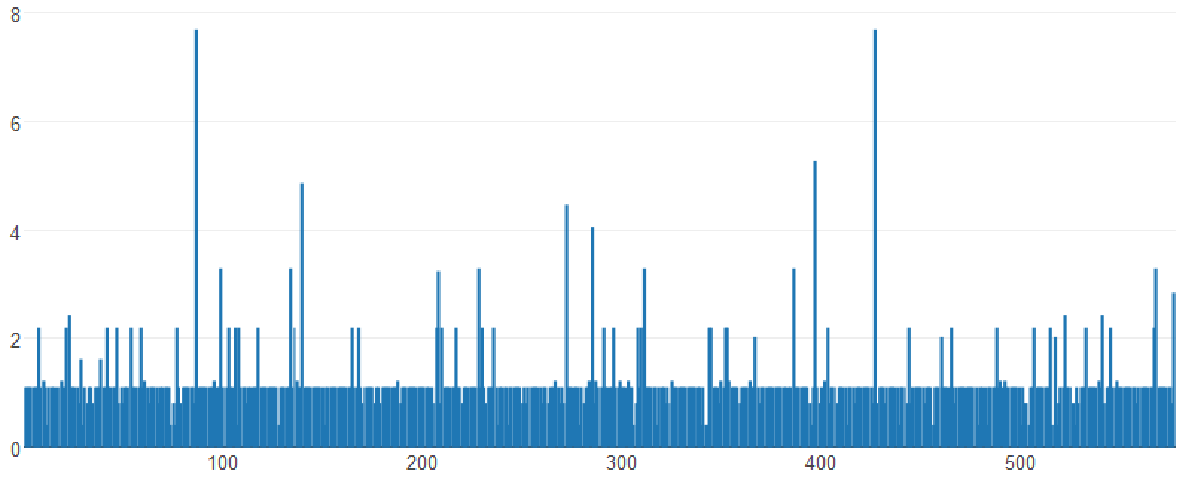

Fig. 3 – Bar plot of tf x idf values for Yemen crisis data

From analyzing the above graphs the minimum and maximum threshold values for tf x idf that are found to be optimal are –

Min tf x idf = 0.9

Max tf x idf = 2.8

b) Cosine Similarity value

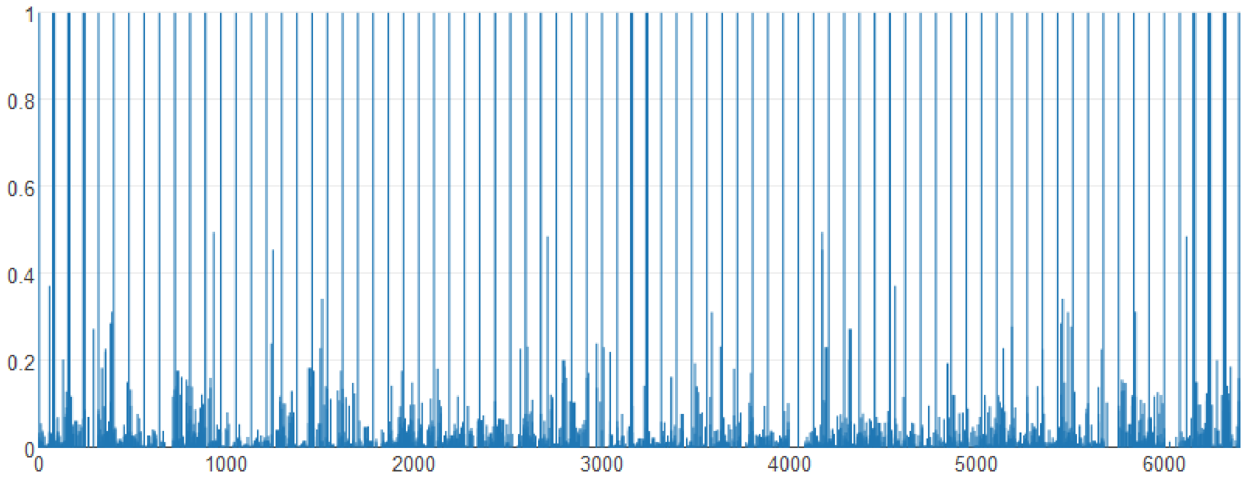

Fig.4 – Bar plot of cosine similarity values for Net Neutrality data

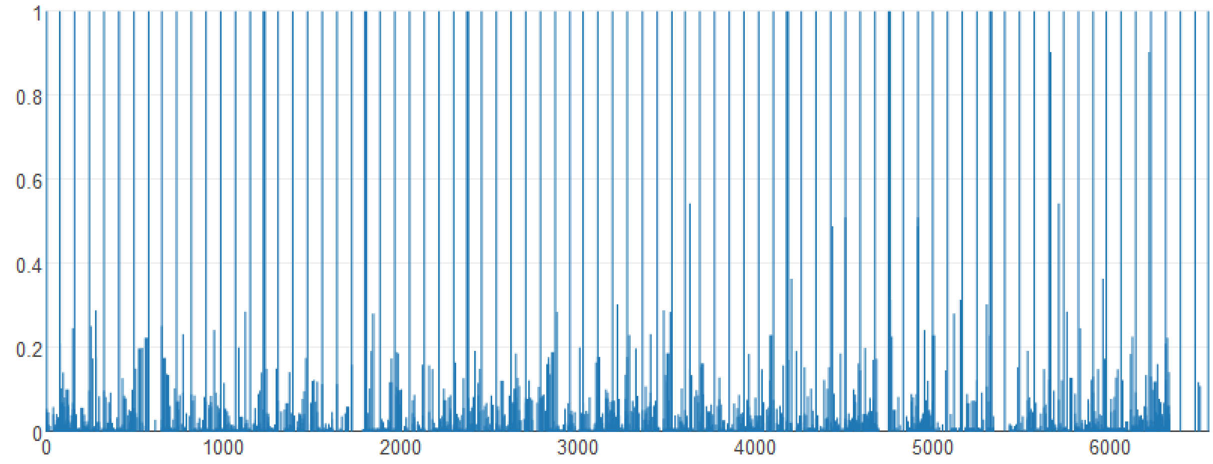

Fig.5 – Bar plot of cosine similarity values for Farmer Suicide data

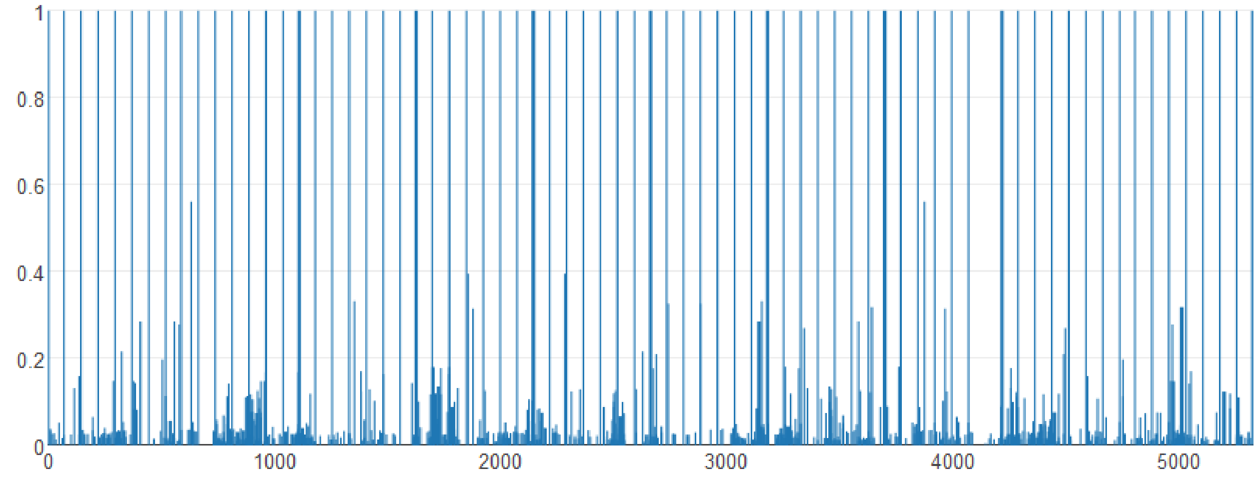

Fig.6 – Bar plot of cosine similarity values Yemen Crisis data

From analyzing the above graphs the optimal value of cosine similarity threshold is found to be –

CosineSimilarityThreshold = 0.05

We also experimented with various values of α (Centroid Based Method ranking weight) and β (Graph Based Method ranking weight) and found that the following values give the best results.

α = 0.5

β = 0.5

Future Plans

1. The hybrid centrality which we were calculating for each sentence was a combination of two different approaches. We can play around by combining more approaches and improve the quality of summarization.
2. The result of our extractive summarization can be used as an input and an abstractive summary can be obtained by implementing abstractive summarization techniques improving the readability, efficiency and understandability.

Conclusion

This report presented a statistical method of generating an extraction based multi-document summary. We used two methods to obtain salient sentences from the cluster of documents. The two methods are centroid based and graph based. The optimal weights of two methods is determined and then the hybrid centrality of each sentence is calculated. Now the sentences having highest hybrid centrality are selected one by one based on the required summary size.

References

1. P. H. Luhn. "Automatic creation of literature abstracts". *IBM Journal*, pp. 159-165, 1958.
2. Daniel'Marcu. "From discourse structures to text summaries", *in Proc. of the ACL'97/EACL'97 Workshop on Intelligent Scalable Text Summarization*, 1997, pp. 82-88.
3. Judith L. Klavans and James Shaw. "Lexical se-mantics in summarization", *in Proc. of the First Annual Workshop of the IFIP Working Group FOR NLP and KR*, Apr. 1995.
4. Simone Teufel and Marc Moens. "Sentence extraction as a classification task", in *Proc. ACL/EACL-97 Workshop on Intelligent Scalable Text Summarization,* 1997, pp. 58-65.
5. Regina Barzilay and Michael Elhadad. "Using lexical chains for text summarization", in *Proc. of the ACL'97/EACL'97 Workshop on Intelligent Scalable Text Summarization*, 1997, pp. 10-17.
6. Jaime G. Carbonell and Jade Goldstein. "The use of MMR, diversity-based reranking for reordering documents and producing summaries", In *Proc. of SIGIR-98*, Melbourne, Australia, 1998.
7. P.B. Baxendale. "Man-made index for technical litterature - an experiment". *IBM J. Res. Dev.*, pp. 354-361, 1958.
8. K. Sparck-Jones. "A statistical interpretation of term specificity and its application in retrieval". *Journal of Documentation*, 1972.
9. Inderjeet Mani and Eric Bloedorn. "Multi-document summarization by graph search and matching". In *Proc. of the Fourteenth National Conference on Artificial Intelligence (AAAI-97)*, 1997, pp. 622-628.
10. Julian Kupiec, Jan O. Pedersen, and Francine Chen. "A trainable document summarizer". *In Research and Development in Information Retrieval*, 1995, pp. 68-73.
11. Dragomir R. Radev, Hongyan Jing, and Malgorzata Budzikowska. "Centroid-based summarization of multiple documents: sentence extraction, utility-based evaluation, and user studies", *In ANLP/NAACL Workshop on Summarization*, Seattle, Apr. 2000.

Remarks and Suggestions