ABSTRACTIVE MULTI-DOCUMENT TEXT SUMMARIZATION USING AUTOMATIC TEXT SUMMARIZER ALGORITHM

S.Sridevi¹, S.Priya²

¹Research Scholar, Department of Computer science, Theivanai Ammal College For Women Villupuram.
²Assistant professor Department of computer Application, Theivanai Ammal College For Women Villupuram

Abstract- The number of web pages on the World Wide Web is increasing very rapidly. Consequently, a search engine like Google, AltaVista, and Bing etc. provides a long list of URLs to the end user. So, it becomes very difficult to review and analyze each web page manually. That’s why automatic text summarization is used to summarize the source text into its shorter version by preserving its information content and overall meaning. In Existing system Automatic Text Summarization algorithm used only Extractive Summarization. Using text summarizer user gets sense of the full-text, or able to know its information content without reading all sentences within the larger text. Text summarization reduces the text by removing less useful data which helps user to find the required information quickly without wasting time in reading the whole text. Lot of work has been done for automatic text summarization. Text summarization is the most challenging task in information retrieval. Data reduction helps a user to find required information quickly without wasting time and effort in reading the whole document collection. In proposed system to use automatic text summarization algorithm a combined approach to document and sentence clustering as an abstractive technique of summarization.

Keywords: Abstractive summarization, Extractive Summarization, web page, similarity, Automatic Text summarizer.

1. INTRODUCTION

Today’s world is full of information, that too online. The World Wide Web contains billions of documents and is growing at an exponential pace, it has become increasingly important to provide improved mechanisms to find and present textual information effective. Text summarization (TS) is the process to reduce (long) textual information to its most essential points; TS systems are designed to take a single article, a cluster of news articles, a broadcast news show, or an email thread as input, and produce a concise and fluent summary of the most important information.

Before discussing TS, first we should know what a summary is. Summary is a document produced from one or more documents, that tell important information from original text and it is shorter than it. Text summarization falls into two categories extractive and abstractive text summarization. Extractive summarization consists of selecting important sentences, paragraphs etc. from the original document and presenting them into shorter form. Extractive summaries are framed by extracting key text segments (sentences or passages) from the text. Abstractive summarization attempts to develop an understanding of the main concepts in a document and then express those concepts in clear natural language. It uses linguistic methods to examine and interpret the text and then to find the new concepts and expressions to best describe it by generating a new shorter text that conveys the most important information from the original text document. This paper presents a new approach with query
and features based multi-document summarization using mutual reinforcement and relevance propagation performance in comparison with only manifold ranking based multi-document summarization and basic like only query based summarization. This motivated us to combine features profile with RARP and query based summarization.

II. RELATED WORK

Query sensitive text summarization technique that can provide the summary of single or multiple web pages was purposed. There user could select a set of links from the search engine results and then text summarizer returned the summary of selected links. Concept based segmentation technique utilized the Document Object Model (DOM) tree to analyze the contents of the web page. The leaf node of this tree was called micro block and adjacent micro block were merged to form a topic block. Each of these sentences were labeled by using ASSERT software. Topic blocks containing information about similar concept word were merged to form a concept block. The results were arranged in descending order of sentence similarity score. The top scoring sentences were extracted and their corresponding web pages were arranged in hierarchical structure. The experimental results proved to be superior in terms of control over the results, quick decision making and reduction of time complexity during processing. But nothing was done on tabular data. Multiple document text summarization technique for improving the effectiveness of retrieval and accessibility of e-learning was purposed. The original document was partitioned into range block and then transformed into a hierarchical tree structure. The range block was represented by nodes of the tree. Then the number of sentences according to the comparison ratio was extracted and some significance score was assigned to them. In traditional summarization techniques; the importance of any sentence was indicated by its location. But today, the textual information like news inside a node was considered equally important regardless of its location inside the node. Therefore, the location feature was not considered during hierarchical summarization of the tree structure. The results of proposed work were tested using t-test and found more superior than the existing system of summarization. Early work in summarization dealt with single document summarization where systems produced a summary of one document. Now a day everyone refers many documents to get more information in less time. Thus study of single document summarization evolved a new type of summarization i.e. multi-document summarization. It was motivated by use of the web. It produced a summary of bundle of documents.

A. Extractive Summarization

Different summarization techniques have been discussed in the literature; for extractive or for abstractive summarization. Extractive summarization assigns a significance score to each sentence and extracts the sentences as it is from the original text(s), with highest scores to form the summaries. The proposed system is based on extractive techniques. Bag-of-words model is built at sentence level, with the usual weighted term frequency and inverse sentence frequency paradigm [2], where sentence-frequency is the number of sentences in the document that contain that term. These sentence vectors are then scored by similarity to the query and the highest scoring sentences are picked to be part of the summary. Tf-idf techniques and clustering of documents is used together to increase the performance. Bundle of documents is related with many topics. They are typically fragmented up into sections. This organization applies even to summaries of documents. Multi-document summarization is also
required clustering of documents related to the topics. The theme is represented by words with top ranking term frequency, inverse document frequency (TF-IDF) scores in that cluster [3]. Once themes have been known, a representative passage in each theme is selected and included in the summary. Different from the traditional query-focused summarization approaches, which were either the simple extensions of generic summarizers and did not uniformly fuse the information in the query and the documents, or based on semi-supervised learning methods and/or supervised learning methods, Wang et al. [7] proposed a manifold-ranking-based approach to make uniform use of sentence-to-sentence and sentence-to-query relationships.

A technique for multi-document text summarization using mutual reinforcement and relevance propagation models was proposed in [10]. It provides the addition of features to sentences with existing query and Reinforcement After Relevance Propagation (RARP). The architecture of RARP consists of three steps i.e. Pre-processing, sentence score calculation based on feature profile and sentence ranking by reinforcement. Pre-processing step consider .txt,.pdf, .rtf, .doc, .html etc. and query as input. Sentence score was calculated using term feature formula. Sentence ranking by RARP and sentence extraction was achieved by using manifold ranking based algorithm. After ranking of sentences, the MDQFS selects the sentences using compression rate of user's choice.

Later Dragomir R. Radev et al. [4] developed a multi-document summarizer, MEAD, which generates summaries using cluster centroid. It summarizes clusters of news articles automatically grouped by a topic detection system. MEAD uses Term Frequency-Inverse Document Frequency (TF-IDF) to calculate the weight for the word / term is used to select salient sentences. A. P. Siva Kumar et al. [5] have discussed query based summarization. The use of TS allows a user to get a sense of the content of full-text, or to know its information content without reading all sentences within the full-text. The sentences containing the query phrases are given higher scores than the ones containing single query words. Then, the sentences with highest scores are incorporated into the output summary. Portions of text may be extracted from different sections or subsections. The resulting summary is the union of such extracts. The number of extracted sentences depends on the compression rate given by users. Feature profile is generated by considering word weight, sentence position, sentence length, sentence centrality, proper noun in the sentence and numerical data in sentence.

Xiaoyan Cai et al. [1] have discussed reinforcement after relevance propagation (RARP) i.e. manifold ranking based relevance propagation with mutual reinforcement between sentences and clusters. They ranked a sentence higher if it is contained in the theme cluster which is more relevant to the given query while a theme cluster ranked higher if it contains many sentences which are more relevant to the given query.

Algorithm

Algorithm: Automatic Text Summarization
Input: User query Q, selection of web pages (WP), downloaded WP.
Output: Summarized web document containing summary of selected document(s).
// Start of algorithm
Step 1. Get the extracted URLs from the SE.
Step 2. Select the URLs for downloading the WP.
Step 3. Collect the downloaded WP in the local repository.
Step 4. Clean the downloaded web pages.
Step 5. Apply the concept based algorithm [7] for each selected document(s).
Step 7. Select the top scoring sentences for summarization.
Step 8. Returned summarized document to the end user.
Step 9. Stop.
B. Manifold Ranking for Multi-Document Summarization

The summary is produced by choosing the sentences with highest overall scores, which are deemed both informative and novel, and highly biased to the given topic. Manifold ranking is a semi-supervised learning that explores the relationship among all the data points in the feature space [7], [8]. It has two versions regarding the different tasks: i) to rank the data points, or ii) to predict the labels of the unlabeled data points. For the task of ranking, the prior assumptions of it include a) nearby points are likely to have the same ranking scores; and b) points on the same structure (typically referred to as a cluster or a manifold) are likely to have the same ranking scores.

C. Mutual Reinforcement for Multi-Document Summarization:

Mutual reinforcement principle that was capable of extracting significant sentences and key phrases at the same time. In his work, a weighted bipartite document graph was constructed by linking together the sentences in a document and the terms appearing in those sentences. The mutual reinforcement was reduced to a solution for the singular vectors of the transition matrix of the bipartite graph. The relevance of each text unit to the given query was calculated by the cosine similarity and characterized by the corresponding text vertex in a three-layer text graph.

III. LITERATURE REVIEW

The approach presented in [3] is to cluster multiple documents by using document clustering approach and to produce cluster wise summary based on feature profile oriented sentence extraction strategy. The related documents are grouped into same cluster using threshold-based document clustering algorithm. Feature profile is generated by considering word weight, sentence position, sentence length, sentence centrality, proper nouns and numerical data in the sentence. Based on the feature profile a sentence score is calculated for each sentence. This system adopts Term Synonym Frequency-Inverse Sentence Frequency (TSF-ISF) for calculating individual word weight. According to different compression rates sentences are extracted from each cluster and ranked in order of importance based on sentence score. Extracted sentences are arranged in chronological order as in original documents and from this, cluster wise summary is generated. The output is a concise cluster-wise summary providing the condensed information of the input documents.

Kamal Sarkar presented an approach to SentenceClustering-based Summarization of Multiple Text Documents in [4]. Here three important factors considered are: (1) clustering sentences (2) cluster ordering (3) selection of representative sentences from the clusters. For the sentence clustering the similarity histogram based incremental clustering method is used. This clustering approach is fully unsupervised & is an incremental dynamic method of building the sentence clusters. The importance of a cluster is measured based on the number of important words it contains. After ordering the clusters in decreasing order of their importance, top n clusters are selected. One representative sentence is selected from each cluster and included in to the summary. Selection of sentences is continued until a predefined summary size is reached.

Pre-processing

The system takes all types of text documents i.e. .txt, .pdf, .rtf, .doc, .html etc. and query as input. Firstly it converts all documents in .txt files. Then it tokenizes the text documents in order to find the individual terms. Then filtering of the text is done by removing the
stop words and remaining words are stemmed using Porter Stemmer algorithm.

**IV. COSINE SIMILARITY**

Cosine similarity is a technique to find out the similarity between pairs of sentences in a document. First of all, the keywords in each pair of sentences i.e. nouns; adjectives are extracted and stored separately. The presence of these words or their meanings in the considered pair of sentences is found out. The absence of the keyword is indicated by 0 and the presence is indicated by the number of occurrences, this account for giving more weightage to a word occurring more than once. Let Pi and Pj be the vectors that indicate the presence of keywords.

\[
P_i = (1, 1, 1, 1, 1, 1, 1, 0)
\]

\[
P_j = (1, 0, 1, 1, 0, 0, 0, 1)
\]

Using the vectors Pi and Pj, the cosine similarity value is calculated using the following formula:

\[
\cos(P_i, P_j) = \frac{P_i \cdot P_j}{|P_i| \cdot |P_j|}
\]

Where, \(P_i \cdot P_j\) is the vector dot product of vectors Pi and Pj.

The dictionary lookup was speeded up using a cache and initiating computations which can go on simultaneously in parallel. This was accomplished by leveraging Multi-Threading in Java.

**V. METHODOLOGY**

This paper proposes a new approach to multi-document summarization. The method ensures good coverage and avoids redundancy. It is the clustering based approach that groups first, the similar documents into clusters & then sentences from every document cluster are clustered into sentence clusters. And best scoring sentences from sentence clusters are selected in to the final summary.

We find similarity between each sentence & query. To find similarity “cosine similarity measure” is used. Given two vectors of attributes, A and B, the cosine similarity, θ, is represented using a dot product and magnitude as:

The important part is generation of vectors. As per above equation both vectors must be of same size because Σ is from 1 to n for both sentence & query. So we have merged sentence & query. Then we take each word from the merged sentence & check whether that word appears in sentence & query both. If yes then we have used the weight (tf*idf) of the word from document & placed that value in vector of sentence for the ith location in vector, & term frequency of the term is placed in vector of query. If the word appears either in sentence or query then the weight of the word is placed in appropriate vector1 & 0 is placed in the vector which doesn’t contain the word.

**VI. EXPERIMENT ANALYSIS**

In order to evaluate the performance of our systems, we use two data sets that have been used in recent multi-document summarization shared tasks: multi-document summarization(task 2) in DUC-2004 and the multilingual multi-document summarization task in MSE-2005. We first show the results of the purely extractive system on each of these tasks, and also show the effects of variations of the systems. Next, we perform experiments using the sentence simplification system, showing additional improvements.

**DUC 2004**

In the multi-document summarization task in DUC-2004, participants are given 50 document clusters, where each cluster has 10 news articles discussing the same topic, and are asked to generate summaries of at most 100 words for each cluster. Since the same task was also held in DUC-2003, but with different documents, we take the 2003 data for
development, especially for training the probabilities. We present the results of our system and SumBasic using different term scoring methods. We also compare them with the best system (peer65) and the baseline system (greedyline) in DUC-2004. As mentioned previously, greedyline simply takes the first 100 words of the most recent news article in the document cluster as the summary. For the evaluation, we use the ROUGE-1 metric (with stemming and stop-words removed), which has been shown to correlate well with human judgments [Lin and Hovy, 2003; Lin, 2004] and which was found to have one of the best correlations with human judgments on the DUC-2004 data [Over and Yen, 2004]. In addition, we also report the performance on ROUGE-2 (bigram overlap) and ROUGE-SU4 (skip bigram) metrics.

VII. CONCLUSION

This paper proposed Query and Features based Multi-Document Summarization using Mutual Reinforcement and Relevance Propagation models to enhance the performance of the existing query based multi-document summarization using RARP algorithm. The proposed MDQFS accepts all types of file formats like: .pdf, .doc, .rtf, .html, etc whereas the existing text summarizer can handle only .html and .txt format of files. We have concentrated on extractive summarization techniques. We have compared the results developed by different extractive techniques by using Performance and correctness measures such as Precision, Recall and F-measure. As per results our novel method outperforms the other methods & it reduces redundancy due to clustering. In future, we would like to improve the system by adding sentence simplification technique for producing summary i.e. it can be used to simplify the sentences which are complex and very large. This approach can also be extended to multi lingual platform. We can also add paraphrasing technique to give abstractive feel to summary.

REFERENCES