A Novel Approach on Query Based Sentence Extraction

B. B. Biswal
Department of Computer Science & Engineering
College of Engineering Bhubaneswar, Odisha, India

Abstract- Whatever scientific inventions or technology Advancement or IT Solutions are made, we always keep in mind the user requirement, user sentiment and user satisfaction also. A lot of work has done in Automatic text summarization, in this paper we have given importance to the users, the user will provide a query and based on that query the summary will be generated. We rank each sentence in the document by assigning a weight value to each word of the sentence and a boost factor is also added to those terms which appear in bold, italic or underlined or any combination of these features and also we assign boost factor to tokens present in user query. It presents an overview of an approach of dynamic summarization; describes the design, implementation, and method of evaluating summaries.

Keywords - Automatic Text Summarization, Sentence Extraction, Boost Factor, Term Weight, Search Engine

I. INTRODUCTION

A web search engine is designed to search for information on the World Wide Web and FTP servers. The search results are generally presented in a list of results often referred to as "search engine results pages". When a user enters a query into a search engine (typically by using keywords), the engine examines its index and provides a listing of best-matching web pages according to its criteria, usually with a short summary containing the document's title and sometimes parts of the text. Most search engines support the use of the Boolean operators AND, OR and NOT to further specify the search query. Boolean operators are for literal searches that allow the user to refine and extend the terms of the search. The engine looks for the words or phrases exactly as entered. There is also concept-based searching where the research involves using statistical analysis on pages containing the words or phrases you search for. As well, natural language queries allow the user to type a question in the same form one would ask it to a human. Most search engines employ methods to rank the results to provide the "best" results first. How a search engine decides which pages are the best matches, and what order the results should be shown in, varies widely from one engine to another. Another important fact about search engine in generation of online summary according to user queries so that the user can decide whether to open the page or not.

The most important task of summarization is to identify the most informative (salient) parts of a text comparatively with the rest. Usually the salient parts are determined on the following assumptions [11]:

- They contain words that are used frequently;
- They contain words that are used in the title and headings;
- They are located at the beginning or end sections;
- They use key phrases which emphasize the importance in text;
- They are the most highly connected with the other parts of text.

A summary [2] can be employed in an indicative way as a pointer to some parts of the original document, or in an informative way to cover all relevant information of the text. In both cases the most important advantage of using a summary is its reduced reading time.

In this paper, section-2 consists of related works, section-3 consists of our recent work, section-4 consists of methodology and the algorithm, section-5 consists of result and discussion and finally section-6 consists of conclusion and future work.

II. RELATED WORK

Automatic text summarization is a technique in which a text is summarized by a computer program. Given a text, its summary (i.e., a non redundant extract from the original text) is returned. According to [13], summarization

techniques can be divided in two groups: those that extract information from the source documents (extraction-based approaches) and those that abstract from the source documents (abstraction-based approaches).

Earliest instances of research on summarizing scientific documents proposed paradigms for extracting salient sentences from text using features like word and phrase frequency (Luhn, 1958),[3] position in the text (Baxendale, 1958) [4]and key phrases (Edmundson, 1969)[5].

Related work (Baxendale, 1958) [4], also done at IBM and published in the same journal, provides early insight on a particular feature helpful in finding salient parts of documents: the sentence position. Towards this goal, the author examined 200 paragraphs to find that in 85% of the paragraphs the topic sentence came as the first one and in 7% of the time it was the last sentence. Thus, a naive but fairly accurate way to select a topic sentence would be to choose one of these two.

Edmundson (1969) [5] describes a system that produces document extracts. His primary contribution was the development of a typical structure for an extractive summarization experiment.

The Trainable Document Summarizer [7] in 1995 performs sentence extracting task, based on a number of weighting heuristics. Following features were used and evaluated:

1. Sentence Length Cut-O Feature: sentences containing less than a pre-specified number of words are not included in the abstract
2. Fixed-Phrase Feature: sentences containing certain cue words and phrases are included
3. Paragraph Feature: this is basically equivalent to Location Method feature in [8]
4. Thematic Word Feature: the most frequent words are defined as thematic words. Sentence scores are functions of the thematic words' frequencies
5. Uppercase Word Feature: upper-case words (with certain obvious exceptions) are treated as thematic words, as well.

In a preliminary work, Boydell used snippets as summary fragments in the field of social Web [8]. Snippets are also used by search engines to provide a textual excerpt of the corresponding Web page according to the keywords used in the query. Snippet can be considered as a topic-driven summarization, since the summary content depends on the preferences of the user and can be accessed via a query, making the final summary focused on a particular topic.

III. OUR RECENT WORK

In this paper we use the extractive method to get the summary of the input document. In order to extract the summary, we use the following features: [12]

1. Content (Key) words: After removing the stop words the remaining words are treated as key words. We have taken the total number of key word during assigning the weight to each term.

2. Frequent key word occurrence in the text: The frequency of the key word which are frequently occurred in the document.

3. Sentence location feature: Usually first sentence of first paragraph of a text document are more important and are having greater chances to be included in summary. So in our case we have made the inclusion of first sentence of the first paragraph of the document is mandatory.

4. Font based features (bold, italic, underlined and their combinations): Sentences containing words appearing in bold, italics or Underlined fonts are usually more important. For this reason we are include this feature in our summarization.

5. Tokens present in the user query are given more importance.

IV. METHODOLOGY

Our summarizer takes input in two formats i.e. .txt and .rtf. Firstly it tokenizes the text in order to find the individual tokens or terms. Then we are filtering the text by removing the stop words. After removing the stop words a weight value is assigned to each individual term. The weight is calculated as follows:

The weight,

$$wt = x * \log\left(\frac{n}{df}\right) + B \quad (1)$$

Where x = Frequency of the Term.
n = Total No. of Sentence exist in the document.
df = No. of sentence contains the Term.
B = Boost factor is added to the term if it is present in user query.

After assigning the weight to each term, the next job is to ranking the individual sentence according to their weight value. The weight of the sentence can be calculated by adding the weight of all the terms in the sentence, i.e.

$$wt_s = \sum_{i=1}^{n}(wt_i) \quad (2)$$

Where wt_s = weight of the sentence.
$wt_1, wt_2, wt_3, \ldots\ldots, wt_n$, are the weights of individual terms in that sentence.
B = Boost factor given to the term if present in user query.

Before ranking the sentence we are adding a boost factor to that term which is appearing in bold, italic, underlined, or any combination of them. Because the term appearing in bold, italic, underlined, or any combination of them, are treated as an important term.

The boost factor is calculated as follows:

$$b = \frac{\text{frequency of the special effect term} * \text{s\_value}}{\text{Total no of special effect term in the document}} \quad (3)$$

Where s_value is taken as follows:
for bold, italic, underlined, s_value=1
for bold-italic, italic- underlined, bold- underlined, s_value=2
for bold-italic-underlined, s_value=3
For a term appears more than once with different special effect, where n is the frequency of that term.

$$s\_value = \sum_{i=1}^{n}(s\_value_i)/n \quad (4)$$

Finally, our summarizer extracts the higher rank sentences including the first sentence of the first paragraph of the document. The number of sentences extracted is based on the user requirement i.e. the percentages of summary the use give as input. This percentage is calculated by dividing the percentage given by the user by total number of ranked sentences, and then taking the ceiling of that result.

Algorithm

Input: A text in .txt or .rtf format.

Output: A relevant summarized text which is shorter than the original text remaining the theme or concept constant.

1. Read a text in .txt or .rtf format and split it into individual tokens.
2. Remove the stop words to filter the text.
3. Assign a weight value to each individual term. The weight is calculated as:
   $wt = \text{Frequency of the Term} * \log\left(\frac{n}{df}\right) + B$
   Where n = Total No. of Sentence exist in the document.
   df = No. of sentence contains the Term.

B = Boost factor is added to the term if it is present in user query.

4. Add a boost factor to that terms which are appear in bold, italic, underlined or any combination of these. The boost Factor can be calculated as:

$$b = \frac{\text{frequency of the special effect term} \times \text{s\_value}}{\text{Total no. of special effect term in the document}}$$

5. Rank the individual sentences according to their weight value as :

$$wt_s = \sum_{i=1}^{n} (wt_i) \qquad (5)$$

Where wt_s = weight of the sentence.

$wt_1, wt_2, wt_3, \ldots\ldots\ldots wt_n$, are the weights of individual terms in that sentence.

6. Finally, extract the higher ranked sentences including the first sentence of the first paragraph of the input text in order to find the required summary. The number of sentences extracted is based on the user requirement i.e. the percentages of summary, the user give as input.

V. RESULT AND DISCUSSION

We have tested our system with 5 documents and 10 queries for each document. Here each document contains around 30 sentences. For auto summarization we have fixed the percentage of summary as 50%, i.e. it will reduce the summary to half of the original document. We have evaluated our system with Kappa measure The Screen shot of our system is given below.

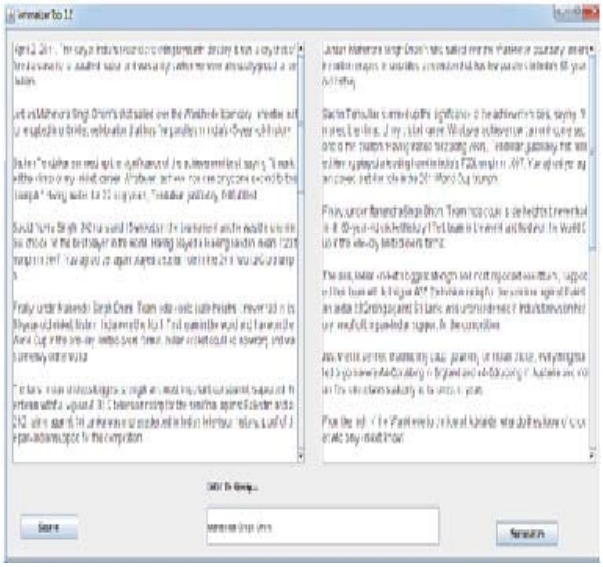

Figure 1. Kappa measure

We tested the system with different documents belong to different domain and each document with 5 different query. Each query produces one summery; So 5 queries produce 5 different summaries. We evaluate the summaries using kappa measure. We are showing kappa measure a document with 5 queries. 5 summaries are verified by 5 pair of reviewers.

Table -1 Different Data Sets

	B (Yes)	B (No)
A(Yes)	3	0
A(No)	1	1

	B (Yes)	B (No)
A(Yes)	4	0
A(No)	0	1

	B (Yes)	B (No)
A(Yes)	3	1
A(No)	0	1

	B (Yes)	B (No)
A(Yes)	2	0
A(No)	1	2

	B (Yes)	B (No)
A(Yes)	4	0
A(No)	0	1

$$k = \frac{Pr(a) - Pr(e)}{1 - Pr(e)} \quad (6)$$

Calculate the value of k for each table
k1=0.54
K2=1.00
K3=0.54
K4=0.61
K5=1.00
Average K = (0.54+1.00+0.56+0.61+1.00)/5 =0.738

The result is satisfactory; we are trying to improve it.

VI. CONCLUSION AND FURURE WORK

In this paper we have improved our result in comparison to our previous work [12]. The font based feature i.e. bold, italic, underlined and all the combination of these are considered to be more important when calculating the weight for ranking the sentences of the document. For this reason the accuracy rate of our system is more than that of Ms-Word automatic text summarization in most cases. Textual Entailment is a NLP task which finds cohesive nature of two sentences. If two sentences are highly cohesive i.e. they are more similar so we should not keep two similar meaning sentences in the summary. This is an important issue in automatic text summarization. We are working to resolve Textual Entailment problem in text summarization.

REFERENCES

[1] Farshad Kyoomarsi, Hamid Khosravi, Esfandiar Eslami and Pooya Khosravyan Dehkordy, "Optimizing Text Summarization Based on Fuzzy Logic", *In proceedings of Seventh IEEE/ACIS International Conference on Computer and Information Science (icis 2008.)*

[2] Computer and Information Science, IEEE, University of Shahid Bahonar Kerman, UK, 347-352, 2008.

[3] Weiguo Fan, Linda Wallace, Stephanie Rich, and Zhongju Zhang, "Tapping into the Power of Text Mining", Journal of ACM, Blacksburg, 2005.

[4] Luhn, H. P. "The automatic creation of literature abstracts". IBM Journal of Research Development, 2(2):159-165, 1958.

[5] Baxendale, P. "Machine-made index for technical literature - an experiment". IBM Journal of Research Development, 2(4):354-361, 1958.

[6] Edmundson, H. P. "New methods in automatic extracting". Journal of the ACM, 16(2):264-285, 1969.

[7] H. P. Edmundson., " New methods in automatic extracting", Journal of the ACM, 16(2):264-285, April 1969.

[8] J. Kupiec, J. Pedersen, and F. Chen, "A trainable document summarizer", In Proceedings of the 18th ACMSIGIR Conference, pages 68-73, 1995.

[9] Ronald Brandow, Karl Mitze, and Lisa F. Rau. "Automatic condensation of electronic publications by sentence selection". Information Processing and Management, 31(5):675-685, 1995.

[10] G Erkan and Dragomir R. Radev, "LexRank: Graph-based Centrality as Salience in Text Summarization", Journal of Artificial Intelligence Research, Re-search, Vol. 22, pp. 457-479 2004.

[11] D.Maru: "From discourse structure to text summaries" in Proceedings of the ACL/EACL '97 Workshop on Intelligent Scalable Text Summarization, pp 82-88, Madrid, Spain.

[12] R.C. Balabantaray, D.K. Sahoo, B. Sahoo, M.Swain, "Text Summarization using Term Weights" IJCA Volume 38-Number 1,pp 10-14, JANUARY-12.

[13] Kolcz, A., Prabakarmurthi, V., Kalita, J. "Summarization as feature selection for text categorization In" CIKM '01: Proceedings of the tenth international conference on Information and knowledge management, pp. 365–370. ACM, New York, NY, USA 2001.