A Novel Approach to Sentence Extraction Using Font Features

R.C. Balabantaray, D. K. Sahoo, B. Sahoo, M. Swain

CLIA Lab, IIIT, Bhubaneswar, Odisha, India

ABSTRACT
As information overload grows day by day, systems that can automatically summarize documents becomes increasingly studied and used. In this paper we have given a novel statistical approach for text summarization of a single source document by sentence extraction using font features. 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. In this paper we improved our result in comparison to our previous paper: [14].

Keywords
Automatic text summarization, sentence extraction, boost factor, term weight

1. INTRODUCTION
In the world of information, the increasing availability of online information has necessitated intensive research in the area of automatic text summarization.

The goal of automatic text summarization is condensing the source text into a shorter version preserving its information content and overall meaning. 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 [13]:

- 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.

Text Summarization methods can be classified into extractive and abstractive summarization. An extractive summarization method consists of selecting important sentences, paragraphs etc. from the original document and concatenating them into shorter form. The importance of sentences is decided based on statistical and linguistic features of sentences. An Abstractive summarization [9][10] 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. In this paper we focus on novel techniques which are based on extractive text summarization methods.

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.

2. RELATED WORK
Summarization task is done in two different methods, i.e. extractive and abstractive. An extractive summarization method consists of selecting important sentences, paragraphs etc. from the original document and concatenating them into shorter form without changing the original concept or meaning of the document. The importance of sentences is decided based on statistical and linguistic features of sentences. An Abstractive summarization [10][11] 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.

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. At first, the author developed a protocol for creating manual extracts that was applied in a set of 400 technical documents. The two features of word frequency and positional importance were incorporated from the previous two works. Two other features were used: the presence of cue words (presence of words like significant, or hardly), and the skeleton of the document (whether the sentence is a title or heading). Weights were attached to each of these features manually to score each sentence. During

Proc. of the International Conference on Advances in Computing and Networking – ACN 2012
Edited by Dr. R. K. Singh.

ISBN: 978-981-07-2683-6

evaluation, it was found that about 44% of the auto-extracts matched the manual extracts.

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.

3. 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: [14]

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.

4. 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,

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

Where x = Frequency of the Term.
n = Total No. of Sentence exist in the document.
df = No. of sentence contains the Term.

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

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:

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

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.

$$\text{s\_value} = \sum_{i=1}^{n} (\text{s\_value}_i) / n \quad (3)$$

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.

4.1 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 terms. The weight is calculated as:
   $$wt = \text{Frequency of the Term} * \log\left(\frac{n}{df}\right)$$
   Where n = Total No. of Sentence exist in the document.
   df = No. of sentence contains the Term.

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

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

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

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 use give as input.

5. RESULT AND DISCUSSION

We have tested our system with 10 documents (five .txt and five .rtf). Here each document contains around 20 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. The Screen shot of our system is given below.

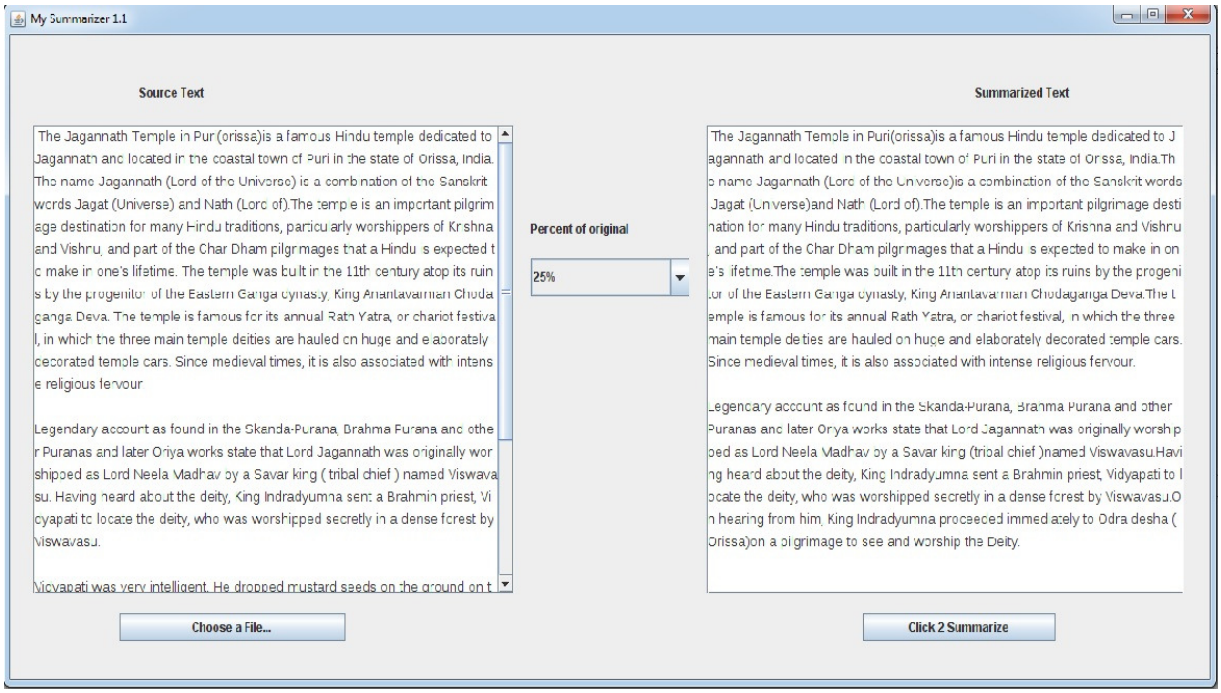

Fig 1: Screen Short of our System

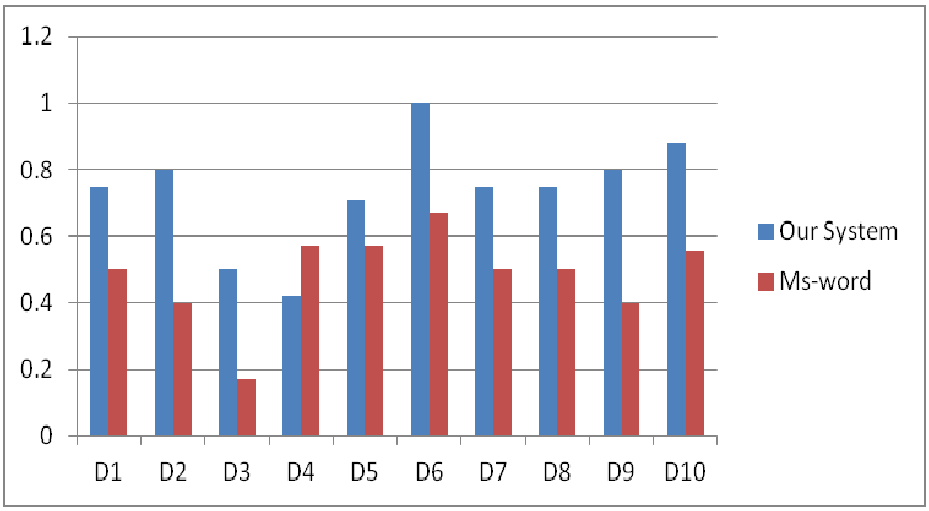

Fig 2: Relevancy of our System and MS-Word w.r.t. Human judgment

A comparison of our system with Ms Word summarization is given in the following table (i.e. Table-1). The relevancy of the summary is calculated with respect to (w.r.t.) human judgment for both the system. The details of the result are given in and the graphical representation of the relevancy of both the system with respect to human judgment is given in figure-2.

Table 1. Result Details

Document Name	No. of Sentences extracted by Ms Word (Sentence No.)	No. of Sentences extracted by our System (Sentence No.)	No. of Sentences extracted by Human Analysis (Sentence No.)	Relevance for Ms Word w.r.t. Human Analysis	Relevance for Our System w.r.t. Human Analysis
Text1.txt	4(2,3,4,5)	4(1,2,4,8)	4(2,4,6,8)	0.5	**0.75**
Text2.txt	5(2,3,4,5,9)	5(1,4,5,6,8)	5(1,2,5,6,8)	0.4	**0.80**
Text3.txt	5(1,4,6,7)	4(1,2,5,7,9,10)	5(1,2,6,8,9,11)	0.17	**0.50**
Text4.txt	5(1,2,3,5,6,10,12)	5(1,2,6,7,9,12,13)	5(1,2,5,7,8,10,11)	**0.57**	0.42
Text5.txt	5(1,2,3,5,6,11)	5(1,2,4,7,11,12,13)	5(1,2,3,7,10,11,13)	0.57	**0.71**
Text6.rtf	3(1,5,6)	3(1,2,5)	3(1,2,5)	0.67	**1.00**
Text7.rtf	4(1,2,3,4)	4(1,3,4,7)	4(1,3,6,7)	0.5	**0.75**
Text8.rtf	4(1,2,7)	5(1,2,5,7)	5(1,5,6,7)	0.5	**0.75**
Text9.rtf	4(1,3,4,10)	4(1,2,4,5,10)	4(1,2,5,6,10)	0.4	**0.80**
Text10.rtf	9(1,5,6,7,10,11,13,14)	9(1,2,7,8,10,11,12,13,17)	9(1,2,5,7,8,10,11,12,17)	0.56	**0.88**

6. CONCLUSION AND FURURE WORK

In this paper we have improved our result in comparison to our previous work [14]. 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.

7. ACKNOWLEDGEMENT

We acknowledge Department of Information Technology (DIT), Ministry of Information & Communication Technology (MCIT), Government of India for this research work.

8. REFERENCE

[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. (1958). The automatic creation of literature abstracts. IBM Journal of Research Development, 2(2):159-165.

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

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

[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] Vishal Gupta, G.S. Lehal, "A Survey of Text Mining Techniques and Applications", Journal of Emerging Technologies in Web Intelligence, VOL. 1, NO. 1, 60-76, AUGUST 2009.

[11] 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.

[12] Udo Hahn and Martin Romacker, "The SYNDIKATE text Knowledge base generator", Proceedings of the first International conference on Human language technology research, Association for Computational Linguistics , ACM, Morristown, NJ, USA , 2001.

[13] 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.

[14] R.C. Balabantaray, D.K. Sahoo, B. Sahoo, M.Swain, "Text Summarization using Term Weights" IJCA Volume 38-Number 1,JANUARY-2012,Pages 10-14.