Parameters Affecting the Judgment Task While Summarizing Documents

Syed Sabir Mohammed[1] and Shanmugasundaram Hariharan[2]
[1]Faculty in Computer Science and Engineering, Sathyabama University, India
[2]Department of Computer Science and Engineering, TRP Engineering College, India

Abstract: *Due to the instability in user agreement, necessity for improvements in document summary generation has paid huge attention. This paper, presents improvements to document summarization by analyzing the sentence position and recommendations of sentences from other sentences for making an efficient document summarizer. As we know humans are good in summarizing the contents, we have made few studies to improve the efficiency of our system much closer to user selection (manual summaries). This article focuses only on providing improvements for news articles. We have attempted to obtain summaries much closer or equal to manually generated summaries and the results obtained were promising. We also, show that term frequency approach combined with position weight gives better results, while adding node weight with the above feature yield results that were significantly better than the former approach. The paper also, illustrates some studies on some common evaluation criterion to generate a unique summary by various users. The results were also, compared with commercially existing Microsoft summarizer. The results produced by us were better as compared to the existing summarizers.*

Keywords: *Single document summarization, stemming, stop words term frequency, sentence extraction.*

Received July 4, 2013; accepted March 17, 2014; published online August 22, 2015

1. Introduction

Summary is defined as the condensed information, which is a shorter version thus preserving the originality of the document. In American Standards Guidelines for Abstracts of American National Standards Institution (ANSI), summary is defined as "An express of a certain document without any explanations and comment" and "It is unnecessary to know who writes the summary". Automatic text summarization is an important and challenging area of Natural Language Processing (NLP). The task of a text summarizer is to produce a synopsis of any document or a set of documents submitted to it. Text summarization is a technique where a computer automatically creates an abstract or summary of one or more texts.

Research on automatic summarizing, taken as including extracting, abstracting, etc., has a long history with an early burst of effort in the sixties following Luhn's pioneering work, two subsequent decades with little research and a low profile and a marked growth of activity since the mid eighties and especially very recently. With the rapid growth of Internet, automatic summarization has attained importance. Lot of researchers started research in area of summarization, which basically started from text, single document, multi document, news, technical paper, audio, video, diagrams and other forms of multimedia.

Reading news online offers many benefits over traditional media. Thousands of news sources are available so, it is chaos for web users to surf all the news sites. With this intent in mind we solely focus on generating high quality summary for the news source.

Internet provides us with new perspectives, making the exchange of information not only easier than ever, but also virtually unrestricted. There is such a vast body of data (news) relating to the event that it is practically impossible to read all of them and decide which are really of interest. A simple visit at, let's say, Google News, Hindu, Indian Express, Deccan Herald and other news services like Sify, Google, Yahoo will show that for certain events with number of hits, i.e., related stories that amounts to millions in fraction of sections. Hence, it is simply impossible to scan through all these documents. A number of commercially available news service provider presents news in the form of individual reports or sometimes as clusters of related articles, allowing readers to easily find all stories on a given topic.

Automatic text summarization is a multi-facetted endeavor that typically branches out in several dimensions. There is no clear-cut path to follow and summarization systems usually tend to fall into several categories at once [21]. Consequently, summarization system falls into at least one; often more than one slot in each of the main categories above and thus must also be evaluated along several dimensions using different measures. A summary can be of a single document or multiple documents [7], generic (author's perspective) or query oriented (user specific) [17], indicative (using keywords indicating the central topics) or informative (content oriented) [8]. A summary can be an extract i.e., certain portions (sentences or phrases) of the text is lifted and reproduced verbatim, whereas producing an abstract involves breaking down of the text into a number of different key ideas, fusion of specific ideas to get more general ones and then generation of new sentences dealing with these new general ideas. In our work we have focused on producing a generic, extractive, informative, single document summary.

Although, some summarizing tools are already available, with the increasing volume of online information, it is becoming harder to generate meaningful and timely summaries. Tools such as Copernic, Microsoft's AutoSummarize, IBM's intelligent text miner, subject search Summarizer and sinope summarizer are useful, but their application is limited, whose function is selecting original pieces from the source document and concatenating them to yield a shorter text. Text extraction is an open approach to summarizing, since there is no prior presumption about what sort of content information is of importance.

In the existing system, job of identifying most important information has become an important issue. The resulting system generated summaries are often not like those summaries written by humans. Linguistic theory tells us that humans are taught to organize text in a particular way, with the overarching structure of the text in mind [18]. Over past few years, data, tasks, techniques and system evaluation procedures have changed a lot. Nevertheless the following issues still remain to be quiet challenging.

- Most manual summaries are clearly better than most automatic summaries.
- Most automatic summaries do differ significantly.

Humans are extremely good in capturing the theme of the document while machines do not achieve the same automatically. So, in our work we made an effort to rank the sentences by our method (by adding up position weight and recommendation weight features) and try to achieve summary as equal to an ideal summary generated by humans.

The rest of the paper is organized as follows. Section 1 presented some introduction on summarization tasks, while comparison of abstraction and extraction is presented compared in section 2. The characteristic of news articles is given in section 3. Section 4 presents the related work. Section 5 gives the experimental results and analysis. Section 6 presents some illustrations on the methodology and improvements to maximize the efficiency of user ranking. Finally, section 7 presents conclusion and future.

2. Related Works

Luhn [12] in his paper states that: 'It is here proposed that the frequency of word occurrence in an article furnishes a useful measurement of word significance. It is further proposed that the relative position within a sentence of words having given values of significance furnish a useful measurement for determining the significance of sentences. The significance factor of a sentence will therefore be based on a combination of these two measurements'. His assumption is that frequency data can be used to extract words and sentences to represent a document.

Edmundson [4] in his paper describes new methods of automatically extracting documents for screening purposes, i.e., the computer selection of sentences having the greatest potential for conveying to the reader the substance of the document. While previous work has focused on one component of sentence significance, namely, the presence of high-frequency content words (key words), the methods describing here also treat three additional components: Pragmatic words (cue words); title and heading words; and structural indicators (sentence location).

Baxendale's [1] work so-called location method uses the position of the sentence in the document as an indication of its importance. His work is based on assumption that important sentences are located in beginning and end of paragraph. He found that so-called "topic sentences" are most likely to occur as either the first (85%) or the last (7%) in a paragraph.

Goldstein *et al.* [5] in his work groups sentence scoring features into seven categories. Frequency-keyword heuristics use the most common content words as indicators of the main themes in the document. Sentences containing these words are scored using functions of their frequency counts. The title-keyword heuristics assumes that important sentences contain content words that are present in the title and major headings of a document. Location heuristics assume that important sentences lie at the beginning and end of a document, in the first and last sentences of paragraphs, and also immediately below section headings. An indicator phrase contains words that are likely to accompany indicative or informative summary material (e.g., "This report…"). A related heuristics involves cue words. These may include two sets of "bonus" and "stigma" words which are positively and negatively correlated with summary sentences. Example bonus words are "greatest" and "significant". Stigma words are exemplified by "hardly" and "impossible".

Yamada *et al.* [23] in his paper proposed a new method of summarizing Japanese news articles. News articles are often described using fixed words or fixed syntactic constructions that are distinctive of the topic. These fixed expressions are considered elements that are essential to the topic. The method generates summary sentences composed of these extracted fixed expressions that include main events without providing preliminary knowledge such as what events are important to the topic.

Mihalcea and Tarau [14] in her work has ranked the sentences in a given text with respect to their importance for the overall understanding of the text. She constructed a graph by adding a vertex for each sentence in the text, and edges between vertices are established using sentence inter-connections. These connections are defined using a similarity relation, where "similarity" is measured as a function of content overlap. Such a relation between two sentences can be seen as a process of "recommendation": A sentence that addresses certain concepts in a text gives the reader a "recommendation" to refer to other sentences in the text that address the same concepts, and therefore a link can be drawn between any two such sentences that share common content. The overlap of two sentences can be determined simply as the number of common tokens between the lexical representations of two sentences, or it can be run through syntactic filters, which only count words of a certain syntactic category.

Svore *et al.* [22] presented a new approach called neural nets called NetSum. Using the RankNet learning algorithm, every sentence in the document is scored and most important sentences are identified. The authors used documents, which consisted of three highlight sentences and the article text. Each highlight sentence is human-generated, but is based on the article. The output of system consists of purely extracted sentences, where there is no sentence compression or sentence generation. The authors have developed two separate problems based on document set. First, can we extract three sentences that best "match" the highlights as a whole? In this task, we concatenate the three sentences produced by our system into a single summary or block, and similarly concatenate the three highlight sentences into a single summary or block and then the system's block is compared against the highlight block. Second, three sentences that best "match" the three highlights were extracted, such that ordering is preserved. Credit is not given for producing three sentences that match the highlights, but are out of order. The second task considers ordering and compares sentences on an individual level, whereas the first task considers the three chosen sentences as a summary or block and disregards sentence order.

3. Extraction Versus Abstraction

Summarization approaches as mentioned above is divided into two groups, text extraction and text abstraction. Text abstraction, being the more challenging task is to parse the original text in a deep linguistic way, interpret the text semantically into a formal representation, find new more concise concepts to describe the text and then generate a new shorter text, an abstract, with the same information content. However, the task of generating a more natural-looking summary is not only difficult but also, unreliable with this state-of-the-art NLP techniques. It is obvious that in order to generate a summary based on text understanding, there should be a meaning representation and a technique for generating a summary from the representation. Constructing a semantic representation of a document, in turn, requires semantic analysis and a knowledge representation technique [10, 13] as well as other lower language processing techniques. In spite of the advantages, this approach suffers from certain disadvantages like, the need to use domain knowledge. As a way to alleviate this problem, some researchers attempt to make use of lexical databases such as WordNet [15] instead of a domain knowledge base.

Text extraction means to identify the most relevant passages in one or more documents, often using standard statistically based information retrieval techniques. This approach generally scores sentences and presenting those with the best scores with or without reordering. In other words extraction is referred to as summarizing by using a limited number of sentences extracted from the original text. In text extraction, where 'what you see is what you get', some of what is on view in the source text is transferred to constitute the summary text. Text extraction is an open approach to summarizing, since there is no prior presumption about what sort of content information is of importance. Extraction-based summarization is still a promising solution especially when speed is concerned. To enhance the coherence of summaries, paragraphs were extracted instead of individual sentences [9, 16]. The main thrust of the extraction approach is to select a few representative sentences from the source document, which are indicative of the contents. Extraction algorithms have a strong tendency to select long sentences from the text. To avoid this sentence selection tendency, sentence length cut-off feature has been used [2, 11].

4. Characteristics of News Articles

News describes report a sequence of events of a particular topic. The first sentence in a news article is called the "lead sentence". Traditionally, previous approaches to automatic text summarization have assumed that the salient parts of a text can be determined by applying one or more of the following assumptions:

- Important sentences in a text contain words that are used frequently.
- Important sentences contain words that are used in the title and section headings.
- Important sentences are located at the beginning or end of the paragraphs.
- Important sentences use bonus words such as "greatest" and "significant" or indicator phrases such as "the main aim of this paper" and "the purpose of this article", while unimportant sentences use stigma words such as "hardly" and "impossible".

Over the years there have been many suggestions as to which low-level features can help determine the importance of a sentence in the context of a source text, such as stochastic measurements for the significance of key words in the sentence [12], its location in the source text [4, 12], connections with other sentences [20] and the presence of cue or indicator phrases or title words [6]. The result of this process is an extract, i.e., a collection of sentences selected verbatim from the text.

Summarization attracts importance based on three different features [2]. The first feature namely Sentence Location Feature, sentences at the beginning and at the end of a paragraph is more likely to contain material useful for summary, because sentences are usually hierarchically organized, with crucial information at the beginning and at the end of paragraphs. The second feature named used is paragraph location feature is similar to sentences here paragraphs are also usually hierarchically organized, with crucial information at the beginning and at the end of paragraphs. The third feature is sentence length feature, where sentences that are too short or too long tend not be included in summaries. A fixed threshold of 5 words for shorter sentences and 15 words for longer sentences is used to decide whether the sentence should be included in summary.

5. Experimental Results and Analysis

We started our experiments by extracting sentences as a function of sentence location for manual summarizations with 10% and 50% summarization ratios carried out by human subjects. This result shows that human subjects tend to extract sentences from the first and the last segments under the condition of 10% summarization ratio, whereas there is no such tendency at 50% summarization ratio. This sentence selection does vary for different summarization ratios as well. Table 2 justifies the above points, as we could see the users have provided higher rank for sentences appearing in top order. We organized the sequence of work into different sections from section 5 to 6. The data corpus and the statistics about the corpus is detailed in section 5.1, section 5.2 explains the concept of user ranking for sentences, section 5.3 details the system description and modules involved in it under various sections. Section 6 provides methodology or improvements to maximize the efficiency of our system much closer to user ranking.

5.1. Corpus Description

Our data corpus consisted of 70 news documents collected from various news sources. Each document was hand-collected in different time intervals. Then, these documents were asked to rank by different judges (procedure is explained in section 5.2). Each document includes the title, timestamp, story highlights and article text. Since, we are talking about single document summarization, we are not worried about time stamp, as each article is ranked separately and so as evaluations. The following observations were made from the corpus collected:

- The document has minimum of 12 sentences and maximum of 38 sentences (there is no minimum and maximum cut-off set for sentence count)
- The total number of sentences in the corpus is 725.

In order that, human evaluation to be done for the corpus we distributed the data source to different judges and asked them to rank the sentences according to the order of importance the user's feel. We splitted up this task into two, where we gave 27 documents (395 sentences) to 4 judges and 26 documents (315 sentences) to 3 judges. The agreement between the judges is very low and the agreement Table 1 is shown below.

Table 1. Human agreement for sentences.

Maximum Number of Judges Used	Number of Sentences Taken for Agreement	No of Sentences Agreed	% of Agreement
4	4	17	5
	3	38	10
	2	198	50
	1	142	36
3	3	6	2
	2	117	38
	1	192	61

5.2. User Ranking for Sentences

In the human evaluation protocol nothing prevents a user from assigning different ranks to the same sentence unit. This poses the problem of which rank to be considered for evaluation. To alleviate this problem we choose judges who were fairly knowledgeable in grasping the contents of the news source. Then, each user is asked to grade the sentence in the order they wish to appear in the summary. Table 3 shown below is a sample illustrating how sentences are ranked by different users (after sentences are ranked based on cummulative scores). We also, infer from Table 3 that sentences appearing in top order are voted by majority of users. Moreover, clearly there is a variation in judge's rankings when we move down the order.

The rank of a sentence is taken to be the rank obtained from cumulative score given by the expert community. Each judge score is taken for ranking the sentences by adding up the score cumulatively as shown in Table 2. For the test case produced in Table 2, all 7 sentences ordered in the original order as they appear in news reports. These reports are then ranked by users as respective columns namely judge1 to judge 4. For each rank a suitable weight is assigned. The cumulative score is obtained by adding up the weights corresponding to the user's rank. By this approach we may not eliminate any of the judge's choice, so that the agreement will not be biased towards a user. Score of each judge's rank is inversely proportional to sentence position. If a sentence is ranked as 1 by all the judges then the cumulative score by the four judges would be 28. For the example in Table 2 the minimum and maximum weight a sentence can have is 1 and 7, while the cummulative score obtained for sentence is 10 and 28 respectively. The possible maximum score a sentence can have is 28 and possible minimum score is 4. Irrespective of the variation in the judges rank the score falls within the minimum and maximum cutoff score.

Table 2. Cummulative scores and rank for judges rating.

Sentence No	Judge 1	Judge 2	Judge 3	Judge 4	Score	Rank
1	1	1	1	1	28	1
2	2	3	2	7	18	2
3	3	2	4	5	18	3
4	7	4	3	6	12	5
5	4	6	5	3	14	4
6	5	5	6	4	12	6
7	6	7	7	2	10	7

Table 3. Cummulative ranking for different documents.

Sentence Number	Doc1	Doc2	Doc3	Doc4	Doc5
1	1	1	1	1	1
2	2	2	2	2	2
3	3	4	3	3	5
4	4	5	4	4	4
5	8	7	5	5	3
6	5	3		6	6
7	6	9		7	8
8	7	8		8	7
9		10		9	9
10		6		10	10

From the survey carried out using the corpus we conclude the following:

- Necessity of more than one model summary although we cannot estimate how many model summaries is required to achieve reliable automated summary evaluation.
- Necessity of more than one evaluation for each summary against each model summary.
- Need to ensure a single rating for each system unit.

Though the general inference is as above, we proposed the cummulative scoring concepts and ranking. Moreover it is quiet harder to find an effective summary. So, we took the sentences ranked by the judges and identified the cumulative score, not leaving any of the judges decision. For each document, one human summary was created as the 'ideal' model summary at each maximum length (100%) and then summary at specified compression rate is chosen from this ideal summary. Then, we proceed our investigating taking this ideal summary for evaluation and for further efficiency improvements.

5.3. System Description

The system designed for summarization of news documents by extraction process performs the following steps. The description of each of the steps is discussed below.

5.3.1. Pre-Processing

The second step is pre-processing of the raw input text. As we could see the documents consist of special characters could even be counted if they weren't ignored. So, the documents are subjected to pre-processing of stage-1 initially in our process, and then followed by stage-2. Newspaper source are generally written in different formats and styles. By format and style we mean news reports are represented in paragraph or sentence format, paragraph with sub-headings, words represented in bold, italic, uppercase and so on. News paper reports are also incorporated with special characters (e.g., :, “, ‘, - , ”, --, [,] etc.,). So, pre-processing of documents is put under two different stages as initial and final which is discussed below in the following sub sections.

- *Stage* 1. Initial Preprocessing:

1. Converting all uppercase letters to lowercase.
2. Eliminating number formats.
3. Eliminating all non-letter characters.
4. Eliminating Honorifics.

- *Stage* 2. Final Preprocessing:

1. Removal of Stop Words: Stop words are frequently occurring, insignificant words that appear in a database record, article or web page. Stop words are an application dependent. They apply to the particular database or application (e.g., searching, summarization). It is commonly assumed that words, which are not members of the noun-verb-adjective classes, should be on stop words lists. When a document is summarized by sentence extraction method we assign weights to all the keywords or tokens in the input document. The process of doing such stop word elimination results in better summary generation. Since we eliminate these stop words unwanted sentences would never climb higher up the order.

   Single characters, common two-character and three-character words, frequently repeated words are typically included in the stop word list to maximize performance of summarization process. We collected a database for stop words that is approximately 500 in number which we use here for summarization task.
2. Applying Porter Stemming Algorithm: Truncation, also called stemming is a technique that allows us to search for various word endings and spellings simultaneously. Stemming algorithms [19] are used in many types of language processing and text analysis systems, and are also widely used in summarization, information retrieval and database search systems. A stemmer is a program determines a stem form of a given word. In other words, generates the morphological root of the word. Terms with a common stem will usually have similar meanings. For the example shown below the common root word is CONNECT.

Connecting, Connected, Connection

The suffix stripping process will reduce the total number of terms in the IR system, and hence reduce the size and complexity of the data in the system, which is advantageous.

5.3.2. Sentence Scoring and Weight age

The sentence extraction algorithm generally applies statistical techniques to generate summary. In our extraction process, sentences are scored based on the term frequency. If the term matches the title words then special weight is given to those terms. We adopted the above approach of giving importance to title terms. Then, each sentence is scored by adding up the scores of each term occurring in the document. Note that, sentences are scored after removal and stop words and stemming the samples. We have not focused features like bold, Italics, Uppercase letters features for special weights. An important aspect that is to be discussed is whether a document should have a title or not. Since, our domain is news genre we must have a title for each news report. For instance we are not focusing on chronology of the reports in this paper, which we leave it for future improvements. Once each sentence is scored those sentences are ranked based on the descending order of weights. If title doesn't exist in document, then the weight of the sentence is calculated using Equation 1.

$$Score = \sum_{i}^{n} F_i S_i \tag{1}$$

Where F_i= Term frequency of specific term in the document, S_i= Sentence counter, n= Number of terms in the document and P_i= Position of the sentence in the document

$$Score = \sum_{i}^{n} (F_i * title_{wt}) S_i \tag{2}$$

Where $title_{wt}$=Title term weight. Sentences are arranges based on the descending order of weights. The single document summarizer process single document at a time and generated summary. The score obtained can be also be normalized to fit within a range (if needed), but we have not normalized the scores in our process.

5.3.3. Sentence Selection

After sentence weight has been calculated the sentences are ranked based on the descending order of sentence weights. In case of ties between sentence score, we arbitrarily pick the sentence that occurs earlier in the document. We have not focused on generating the summary on a specific order. Sentences are selected based on the user specified compression ratio. We have picked up sentences at 30%, 40%, 50%, 60% and 70% compression rates and analyzed the results under different study discussed in section 6.

5.3.4. Evaluation

Having calculated the sentence weight, the final step is the efficiency calculation of our system compared to human generated summary. By efficiency we mean how well ours system behaves with that of system in selecting the sentences. We proposed two modes of evaluation or efficiency calculation of our summary.

- By Sentence Selection Method.
- By Judge Score Method.

The sentence selection method calculates the efficiency of summary by picking up the sentences depending on the user specified compression ratio. On contrary efficiency for judge score method is calculated based on the judge score. In this method we call the cumulative score calculated at a specified compression rate as official score and the rank sorted out for the official score as official rank. Table 4 tabulates the judge score, judge rank and rank given by our system (depending on the sentence score arranged in descending order of weights). The efficiency calculation by method 1 and method 2 for 30 % compression ratio is discussed subsequently.

Table 4. Score provided by judges and rank generated.

Sentence Number	Judge Score	Judge Rank	Rank Given by our System
1	41	2	8
2	36	3	4
3	30	5	1
4	42	1	2
5	31	4	5
6	20	6	6
7	18	7	3
8	12	10	10
9	16	8	7
10	13	9	9
11	5	11	11

- **Method 1**. *Judge Score Method*: In sentence selection method for 30% compression rate, 4 sentences are to be picked up. The official sentences selected for inclusion in summary is sentence 4, 1, 2 and sentence 5. The judge's score is added cumulatively to obtain the official score. By our method we select sentences 3, 4, 7 and sentence 2. Efficiency of the system is calculated by dividing the score obtained by our system with the official score.

  Official score= 42+41+36+31= 150
  Our score= 30+42+18+36= 126
  Efficiency= 126 /150= 84 %

- **Method 2**. *Sentence Selection Method*: In sentence selection method for 30% compression rate, the official sentences to be picked up are sentence 4, 1, 2 and sentence 5. The sentences picked up by our system are sentences 3, 4, 7 and sentence 2. The efficiency by the number sentences that matches our sentences selected by our system with that of ideal summary. The efficiency at 30 % for the example illustrated is 50% (since only two sentences are matches the ideal summary). Though both the methods perform some sort of sentence selection procedure, both have differences in efficiency calculation as well sentence calculation.

For Method 1 sentences are selected at the specified compression rate, even though the sentences do not match with that of ideal summary. This is in contrast to Method 2, where sentences that exactly match the ideal summary alone are taken for efficiency calculation.

Table 5 shows sample for comparison of efficiency of calculation by judge score method and sentence selection method. We infer from Table 5 that Method 1 produces higher efficiency, since alternative sentences or sentences that are at next highest rank with that of the picked up sentence is selected for summary. Method 2 least bothers about the number of sentences selected at specified compression rates. Therefore, we conclude that efficiency calculation depends on the necessity of summarization task.

Table 5. Efficiency calculation by judge score method and sentence selection method.

Efficiency/ Weight Multiplied	Efficiency by Judge Score Method					Efficiency by Sentence Selection Method				
	30%	40%	50%	60%	70%	30%	40%	50%	60%	70%
1	0.84	0.90	0.84	0.84	0.89	0.50	0.66	0.57	0.66	0.80
10	0.84	0.90	0.90	0.84	0.89	0.50	0.66	0.57	0.66	0.80
20	0.92	0.90	0.90	0.88	0.89	0.75	0.66	0.57	0.77	0.80
30	0.92	0.90	0.90	0.90	0.89	0.75	0.66	0.57	0.77	0.80
40	0.92	0.91	0.90	0.90	0.89	0.75	0.66	0.57	0.77	0.80
51	0.92	0.91	0.92	0.90	0.89	0.75	0.66	0.71	0.77	0.80
100	0.92	0.89	0.92	0.90	0.96	0.75	0.50	0.71	0.77	0.90

6. Methodology/Improvements to Maximize Efficiency of User Ranking

Our objective of this paper solely lies to make some improvements to existing summary, so that our system equally competes with manually generated system. We have investigated our study by three ways. Each of these studies is listed and discussed further.

6.1. Study 1: Term Frequency Combined with Position Weight

Position plays a vital role in the newspaper documents. Lot of works done earlier for news genre says that most salient points for inclusion in summary can be picked up from top *n* sentences or from lower order. Lot of research been carried in news genre and they all agree upon this issue. The corpus for our experiments consisted only sentences and there is no paragraph feature. So, we have not given any special weights for paragraph feature. The sentence position is calculated using Equation 3.

$$Pos_{wt} = n - i + 1 / n \qquad (3)$$

Where *i* denote position counter and *n* denotes number of sentences.

Therefore, if we have 10 sentences in a document, first sentence has a weight of 1 and second has 0.9 and the last sentence has a weight of 0.1. Using Equations 1 or 2, sentence score is obtained. The modified sentence score for each sentence after adding up the position weight is calculated using Equation 4.

$$Sent_{score} = Score + Pos_{wt} * \lambda_1 \qquad (4)$$

Where λ_1= linear weight multiplier.

Table 6 shows the effect of position weight, when λ_1 is varied. For a linear position weight of 1 to the score equal to the maximum score (we call this as cut-off point) obtained using Equation 2 is shown in Table 6 and graph is shown in Figure 1.

Table 6. Efficiency at different compression rates and modification of position weights.

Efficiency/ Liner Weight	Compression Rates				
	30%	40%	50%	60%	70%
$\lambda_{1=1}$	0.33	0.50	0.60	0.66	0.71
$\lambda_{1=10}$	0.33	0.50	0.40	0.66	0.85
$\lambda_{1=20}$	0.66	0.50	0.40	0.40	0.85
$\lambda_{1=30}$	0.66	0.50	0.40	0.66	0.85
$\lambda_{1=39}$(max)	0.66	0.50	0.40	0.66	0.85
$\lambda_{1=50}$	0.66	0.50	0.40	0.66	0.85

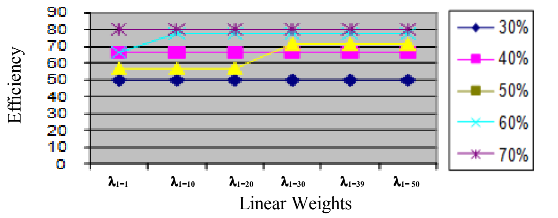

Figure 1. Linear weights vs efficiency of ranking.

Linear weight of 0 ideally means that it is term frequency alone with no position weight, while value of 1 denotes the position weight score combined with sentence score. We have also carried our experimenting our study modifying the liner weight beyond our cut-off feature as discussed earlier. We found that beyond this point the curve saturates and we were not able to improve the efficiency (inferred from Figure 1). For few test cases we found that efficiency that is better at specified rates degrades by altering λ_1. Nevertheless we were able to obtain better efficiency for certain other compression rates. From study 1 we conclude the following:

- For maximizing the efficiency, weight for λ_1 would be the maximum score of the sentence.
- Efficiency at certain specified ratio increases significantly.

6.2. Study 2: Term Frequency Combined with Recommendation Weight

Recommendation weight is a way of deciding on the importance of a vertex within a graph, by taking into account global information recursively computed from the entire graph, rather than relying only on local vertex-specific information. The idea implemented by the ranking model is that of "voting" or "recommendation". When one vertex links to another one, it is basically casting a vote for that other vertex. The higher the number of votes that are cast for a vertex, the higher the importance of the vertex.

We calculate the recommendation weight using similarity existing between the sentences. The overlap of two sentences can be determined simply as the number of common tokens between the two sentences. For convenience we use a normalization factor and divide the content overlap value of two sentences with the maximum of the recommendation weight (so that, all values fall within the range of 0 to 1). The formula for calculating the score is given below:

$$Sent_{score} = Score + Node_{wt} * \lambda_2 \tag{5}$$

Where λ_2= linear recommendation weight.

Using Equation 5 we obtain the score for each sentence and the efficiency is calculated by altering the value for recommendation weight. Table 7 shows the effect of recommendation weight, when λ_2 is varied. The value for the multiplier is varied as we did for study 1 and is shown in Table 7 and the graph is shown in Figure 2. From the study 2 we infer that:

- For maximizing the efficiency, weight for λ_2 would be the maximum score of the sentence, but the results were lesser than the previous approach.
- Efficiency at certain specified ratio increases significantly but it is not inferior then previous approach.

Table 7. Efficiency at different compression rates and linear weights.

Efficiency/ Liner Weight	Compression Rates				
	30%	40%	50%	60%	70%
$\lambda_{2=1}$	0.33	0.50	0.60	0.66	0.71
$\lambda_{2=}$ 10	0.33	0.50	0.40	0.66	0.71
$\lambda_{2=}$ 20	0.33	0.50	0.40	0.66	0.71
$\lambda_{2=}$ 30	0.33	0.50	0.40	0.66	0.85
$\lambda_{2=}$ 39(max)	0.33	0.50	0.40	0.66	0.85
λ_2= 50	0.33	0.50	0.40	0.66	0.85

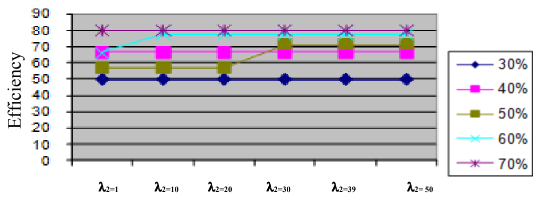

Figure 2. Linear weights vs efficiency of ranking.

6.3. Study 3: Term Frequency Combined with Position Weight and Node Weight

A combination of sentence position and node weight behaves well, which is shown below in Table 8. Using Equation 6 the scores are efficiency variation is shown in Figure 3.

$$Sent_{score} = Sent + Pos_{wt} * \lambda_1 + Node_{wt} * \lambda_2 \tag{6}$$

Table 8. Efficiency at different compression rates and linear weights.

Efficiency/ Weight	30%	40%	50%	60%	70%
λ1, λ2 =10	0.33	0.50	0.40	0.66	0.85
λ1, λ2 =20	0.33	0.50	0.40	0.66	0.85
λ1, λ2 =30	0.66	0.50	0.40	0.66	0.85
λ1, λ2 =39	0.66	0.50	0.40	0.66	0.71
λ1, λ2 =50	0.66	0.50	0.40	0.66	0.71
TF+ Poswt *4max+ Nodewt *λ2	1	0.75	0.60	0.66	0.85

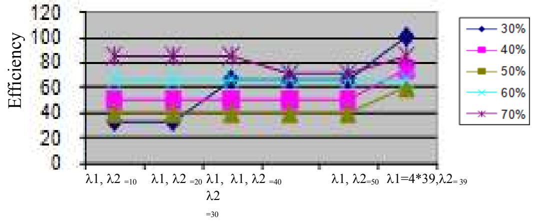

Figure 3. Linear weights vs efficiency of ranking.

6.4. Study Analysis

We have made a study by experimenting the variations in user ranking of news source. From the three studies we made we found that third approach behaves well in improving the efficiency of the system. This is also, inferred from the graph shown in Figure 4. At each specified compression rate there is increase in efficiency for study 3, while study 1 is inferior to study 3 and study 2 is least of all the three approaches.

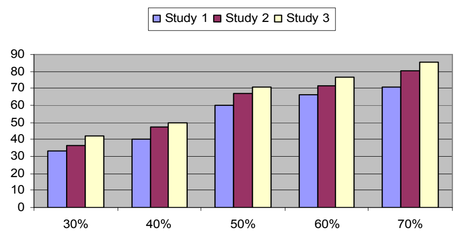

Figure 4. Performance of study results compared with three different studies.

The efficiency of our system is also compared with Microsoft Auto Summarizer. We calculated the efficiency at 30 %, 40 %, 50 %, 60 % and 70 % compression rates respectively. Table 9 shows the comparison of our method with the commercial summarizers. Our results are better compared to any of the studies we made.

Table 9. Efficiency of our proposed system compared to human summary.

Target Rate	Study 1	Study 2	Study 3	Microsoft
30%	0.33	0.36	0.42	0.33
40%	0.40	0.47	0.50	0.25
50%	0.60	0.67	0.70	0.40
60%	0.66	0.71	0.76	0.50
70%	0.71	0.80	0.85	0.87

From the study 3 we found that the value for λ_1 would be four times the maxscore and value is maxscore itself. Study 3 is to improve the efficiency of the system much better than the two previous approaches. So, we combined both studies 1 and 2 by adding up the sentence score with position weight as well as recommendation weight. For better numerical understanding this can be observed from Tables 7 and 8 respectively.

7. Conclusions and Future Improvements

We presented a single document news summarizer that picks up sentences, which incline with user's ranking.

Many of the summarizers at present picks up sentences which deviate by a large margin from manually generated summary. So, an effort has been made to adjust our ranking with the user ranking. We also investigated the effect of position weight and recommendation weight in sentence extraction process. We conclude form the study that Term frequency approach combined with position weight and recommendation weight is an good approach for achieving a good single document summarizer.

With efforts made at single document summary generation, we now focus on developing multi-document summary generation by adopting a language independent summarizer by adopting recommendation weights. We now try to build multi document summary based on these ranks. We have made few attempts to summarize multiple documents of the same cluster. We also, try attempting to cluster documents using similarity functions.

References

[1] Baxendale P., "Machine-Made Index For Technical Literature-An Experiment," IBM Journal of Research and Development, vol. 2, pp. 354-361, 1958.

[2] Chen F., Han K., and Chen G., "An Approach to Sentence Selection Based Text Summarization," in Proceedings of IEEE TENCON, 2002.

[3] Chuang W. and Yang J., "Extracting Sentence Segments for Text Summarization: A Machine Learning Approach," in Proceedings of ACM SIGIR conference, 2000.

[4] Edmundson H., "New Methods in Automatic Extracting," Journal of the ACM, vol .16, no. 2, pp. 264-285, 1969.

[5] Goldstein J., Kantrowitz M., Mittal V., and Carbonell J., "Summarizing Text Documents: Sentence Selection and Evaluation Metrics," ACM SIGIR, pp. 121-128, 1999.

[6] Halliday M. and Hasan R., "Cohesion in English," Longmans, London, 1996.

[7] Hariharan S., Ramkumar T., and Srinivasan R., "Enhanced Graph Based Approach for Multi Document Summarization," The International Arab Journal of Information Technology, vol. 10, no. 4, pp. 334-341, 2013.

[8] Hilda H., "Cross-Document Summarization by Concept Classification," in Proceedings of the 25th annual international ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 121-128, 2002.

[9] Hovy E. and Lin C., "Automated Text Summarization in SUMMARIST," in Proceedings of the Workshop on Intelligent Text Summarization, Madrid, Spain, 1997.

[10] Jacobs P. and Rau L., "Natural Language Techniques for Intelligent Information Retrieval," *in Proceedings of the Eleventh Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, France, pp. 85- 99, 1988.

[11] Jing H., Barzilay R., McKeown K., and Elhadad M., "Summarization Evaluation Methods: Experiments and Analysis," *in Working Notes, AAAI Spring Symposium on Intelligent Text Summarization*, Stanford, CA, 1998.

[12] Luhn H.,"The Automatic Creation of Literature Abstracts," IBM Journal of Research Development, vol. 2, no. 2, pp. 159-165, 1958.

[13] Mckeown K. and Radev D., "Generating Summaries of Multiple News Articles," in Proceedings of the Eighteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 152-161, 1995.

[14] Mihalcea R., and Tarau P., "An Algorithm for Language Independent Single and Multiple Document Summarization," in Proceedings of the International Joint Conference on Natural Language Processing, Korea, October 2005.

[15] Miller G., George A., Beckwith R., Fellbaum C., Gross D., and Miller J., "Introduction to WordNet: an Online Lexical Database," International Journal of Lexicography, vol. 3, no. 4, pp. 235-312, 1990.

[16] Mitra M., Amit S., and Chris B., "Automatic Text Summarization by Paragraph Extraction," in ACL/EACL-97 Workshop on Intelligent Scalable Text Summarization, Madrid, Spain, pp. 31-36, 1997.

[17] Paice C., "Constructing Literature Abstracts by Computer: Techniques and Prospects," Information Processing and Management, vol. 26, pp. 171-186, 1990.

[18] Paice C., "The Automatic Generation of Literature Abstracts: An Approach Based on Self- Indicating Phrases," in Proceedings of the Joint ACM/ BCS Symposium in Information Storage and Retrieval, pp .172-191, 1980.

[19] Porter M., "An Algorithm for Suffix Stripping," Program, vol. 14, no. 3, pp. 130-137, 1980.

[20] Sparck-Jones K., "Automatic Summarizing: Factors and Directions," Advances in Automatic Text Summarization, MIT Press, 1999.

[21] Skorokhodko E., "Adaptive Method of Automatic Abstracting and Indexing," in IFIP Congress, Ljubljana, Yugoslavia, pp. 1179-1182, 1971.

[22] Svore K., Vanderwende L., and Burges C., "Enhancing Single-document Summarization by Combining RankNet and Third-party Sources," in Proceedings of the Joint Conference on Empirical

Methods in Natural Language Processing and Computational Natural Language Learning, pp. 448-457, 2007.
[23] Yamada I., Shibata M., and Kim Y., "Multiple Text Summarization using Fixed Expressions in News Articles," in IEEE, SMC, MP2E4, 2002.

Syed Sabir Mohamed was born in 1981,Trichy, Tamilnadu, India. He completed his B.E (Computer Science and Engineering) at Royal College of Engineering and Technology, Anna University, Chennai. He was awarded M.E (Computer Science and Engineering) from Anna University Tiruchirappalli,Tamilnadu. He has 6 years of experience in teaching both UG and PG programmes. Currently he is presently working as an Assistant Professor in the Department of Computer science and Engineering in M.A.R College of Engineering and Technology, Tamilnadu, India. So far he has guided 8 U.G and 6 P.G students under his supervision.

Shanmugasundaram Hariharan received his B.E degree specialized in Computer Science and Engineering from Madurai Kammaraj University, Madurai, India in 2002, M.E degree specialized in the field of Computer Science and Engineering from Anna University, Chennai, India in 2004. He holds his Ph.D degree in the area of Information Retrieval from Anna University, Chennai, India. He is a member of IAENG, IACSIT, ISTE, CSTA and has 9 years of experience in teaching. Currently he is working as Associate Professor in Department of Computer Science and Engineering, TRP Engineering College, India. His research interests include Information Retrieval, Data mining, Opinion Mining, Web mining. He has to his credit several papers in referred journals and conferences. He also serves as editorial board member and as program committee member for several international journals and conferences.