A Report

On

Document Summarization

Submitted By

Darsh Ambaliya (D18CE161)

Gaurang Ganatra (D18CE164)

Neel Thacker (D18CE166)

Department of Computer Engineering

CSPIT, CHARUSAT

Changa, INDIA

Accredited with Grade A by NAAC

Accredited with Grade A by KCG

Under supervision of

Dhaval Bhoi

Assistant Professor

U & P U. Patel Department of Computer Engineering,

CHARUSAT, Changa, Gujarat.

Introduction:

The intention of text summarization is to express the content of a document in a more concise form that meets the needs of the user. Far more information than can realistically be digested is available on the World-Wide Web and in other electronic forms. News information, biographical information, minutes of meetings missed, it isn't possible to read everything one would want to read and so some form of information shortening is needed. Secondly, the language of news media may be impenetrable to some people, for example, children or people learning English as a foreign language. Some method of language simplification would be useful, as well as a method of providing the background knowledge adults take for granted. A solution that addressed both of those problems would enable a wider range of people to be aware of a greater amount of information.

Potential applications:

- Summarizing medical data for doctors. In the authors intend to summarize medical information with the patients details in mind.
- Multimedia News Summarization This comprise summarizing data from different sources.
- Producing Intelligence Reports Given a wide range of documents, an intelligence analyst may wish to read a life story of a person. A system exists which creates a dossier of information on a person from a text collection.
- Text for Hand-held devices Due to the limited size of displays on WAP phones and palm-top computers, it is useful to condense text found in web pages browsed.
- Convenient Text-to-Speech for Blind people. The idea here is to scan in a page from a book, and then read out a summary of the page rather than the entire text.
- Collating Search Engine hits Rather than read all the pages returned by a search, it would be better to read a summary of the top N hits.
- Summarizing Meetings Combining summarization with automatic speech recognition produces a system which summarizes the important points of a meeting.

Scope:

The project is wide in scope, all of the constraint stated below may seem to contradict that, but they are the only restrictions applied. This project looks at single document summarization the area of multi document summarization is not covered. Also, the summaries produced are largely extracts of the document being summarized, rather than newly generated abstracts. The parameters used are optimum for news articles, although that can be changed easily. With regard to language simplification, only lexical changes were considered syntactic changes were not. Background information was limited to biographical information and maps.

Summarization:

The phases that an automatic summarizer goes through can be split into the following:

Interpret: This is where a characterization of the document to be summarized is produced. Also known as analysis.

Transform: This is where the characterization of the document is turned into one of a summary of the document.

Generate: Here, summary text is produced from the summary representation. Also known as union.

Types of summary:

Summaries of text can be divided into different categories, some of them harder to automate than others. One division is based on the origin of the text in the summary:

Extractive: This where the summary comprises of sentences that have already appeared in the text.

Abstractive: Here, some new text is generated by the summarizer.

Clearly, extractive summaries are the simpler option of the two, since they avoid the language generation problem.

Summaries can also be categorized by their purpose:

Indicative: These summaries are meant to give the reader an idea as to whether it would be useful reading the entire document. The topic and scope of the text should be expressed but not necessarily all of the factual content.

Informative: This type of summary expresses the important factual content of the text.

Critical: This sort of summary criticizes the document. It expresses an opinion on in the case of a scientific paper, say the methods employed and the validity of the results.

Indicative summaries are the most suitable to automate, out of the three, and critical summaries probably the least. Informative summaries are a little harder than indicative ones, since comprehensive coverage of the information in the text is required.

Types of evaluation:

Evaluation of summaries is an analytical problem in this area. Evaluations fall in to one of two types:

Intrinsic: This is where the system is tested in of itself. Typically tests are done to measure summary quality and informativeness.

Extrinsic: This is when the summarization system is measured relative to a real world task. Examples include reading comprehension tests how much of the original content did the user gain from just reading the summary?

Text summarization approaches in literature:

There are various text summarization approaches in literature. Most of them are based on extraction of important sentences from the input text. The first study on summarization, which was conducted by in 1958, was based on frequency of the words in a document. After this study other techniques arose, based on simple features like terms from keywords/key phrases, terms from user queries, frequency of words, and position of words/sentences. The algorithms belonging to Baxendale and Edmundson are examples of the techniques based on simple features.

Statistical methods are another approach for summarization. The SUMMARIST project is a well-known text summarization project that uses a statistical approach. In this project, concept relevance information extracted from dictionaries and WordNet is used together with natural language-processing methods. Another summarization application based on statistics belongs to Kupiec et al. where a Bayesian classifier is used for sentence extraction.

Text connectivity is another approach for dealing with problems of referencing to the already mentioned parts of a document. Lexical chains method is a well-known algorithm that uses text connectivity. In this approach, semantic relations of words are extracted using dictionaries and WordNet. Lexical chains are constructed and used for extracting important sentences in a document, using semantic relations.

There are graph-based summarization approaches for text summarization. As stated in Jezek and Steinberger, the well-known graph-based algorithms HITS and Google's PageRank were developed to understand the structure of the Web. These methods are then used in text summarization, where nodes represent the sentences, and the edges represent the similarity among the sentences. TextRank and Cluster LexRank are two methods that use a graph-based approach for document summarization.

There are also text summarization algorithms based on machine learning. These algorithms use techniques like Naive-Bayes, Decision Trees, Hidden Markov Model, Log–linear Models, and Neural Networks. More detailed information related to machine learning based text summarization approaches can be found in Das and Martins.

In recent years, algebraic methods such as LSA, Non-negative Matrix Factorization (NMF) and Semi-discrete Matrix Decomposition have been used for document summarization. Among these algorithms the best known is LSA, which is based on singular value decomposition (SVD). Similarity among sentences and similarity among words are extracted in this algorithm. Other than summarization, the LSA algorithm is also used for document clustering and information filtering.

Text summarization using LSA:

The algorithms in the literature that use LSA for text summarization perform differently. In this section, information on LSA will be given and these approaches will then be explored in more detail.

Latent Semantic Analysis:

Latent Semantic Analysis is an algebraic-statistical method that extracts hidden semantic structures of words and sentences. It is an unsupervised approach that does not need any training or external knowledge. LSA uses the context of the input document and extracts information such as which words are used together and which common words are seen in different sentences. A high number of common words among sentences specifys that the sentences are semantically related. The meaning of a sentence is decided using the words it contains, and meanings of words are decided using the sentences that contains the words. Singular Value Decomposition, an algebraic method, is used to find out the interrelations between sentences and words. Besides having the capability of modelling relationships between words and sentences, SVD has the capability of noise reduction, which helps to improve accuracy. In order to see how LSA can represent the meanings of words and sentences the following example is given:

Example 1: Three sentences are given as an input to LSA.

d0: ‘The man walked the dog’.

d1: ‘The man took the dog to the park’.

d2: ‘The dog went to the park’.

After performing the calculations we get the following figure, Figure 1. From Figure 1, we can see that d1 is more related to d2 than d0; and the word ‘walked’ is related to the word ‘man’ but not so much related to the word ‘park’. These kinds of analysis can be made by using LSA and input data, without any external knowledge. The summarization algorithms that are based on LSA method usually contain three main steps.

Step 1:

Input matrix creation: an input document needs to be displayed in a way that enables a computer to understand and perform calculations on it. This representation is usually a matrix representation where columns are sentences and rows are words/phrases. The cells are used to display the importance of words in sentences. Different approaches can be used for filling out the cell values. Since all words are not seen in all sentences, most of the time the created matrix is sparse.

The way in which an input matrix is formed is very important for summarization, since it affects the resulting matrices calculated with SVD. As already introduced, SVD is a complex algorithm and its complexity increases with the size of input matrix, which degrades the performance. In order to reduce the matrix size, rows of the matrix, i.e. the words, can be reduced by approaches like removing stop words, using the roots of words only, using phrases instead of words and so on. Also, cell values of matrix can change the results of SVD. There are different approaches to filling out the cell values. These approaches are as follows.

- **Frequency of word:** the cell is filled in with the frequency of the word in the sentence.
- **Binary representation:** the cell is filled in with 0/1 depending on the existence of a word in the sentence.
- **Tf-idf (Term Frequency-Inverse Document Frequency):** the cell is filled in with the tf-idf value of the word. A higher tf-idf value means that the word is more frequent in the sentence but less frequent in the whole document. A higher value also indicates that the word is much more representative for that sentence than others.
- **Log entropy:** the cell is filled in with the log-entropy value of the word, which gives information on how informative the word is in the sentence.

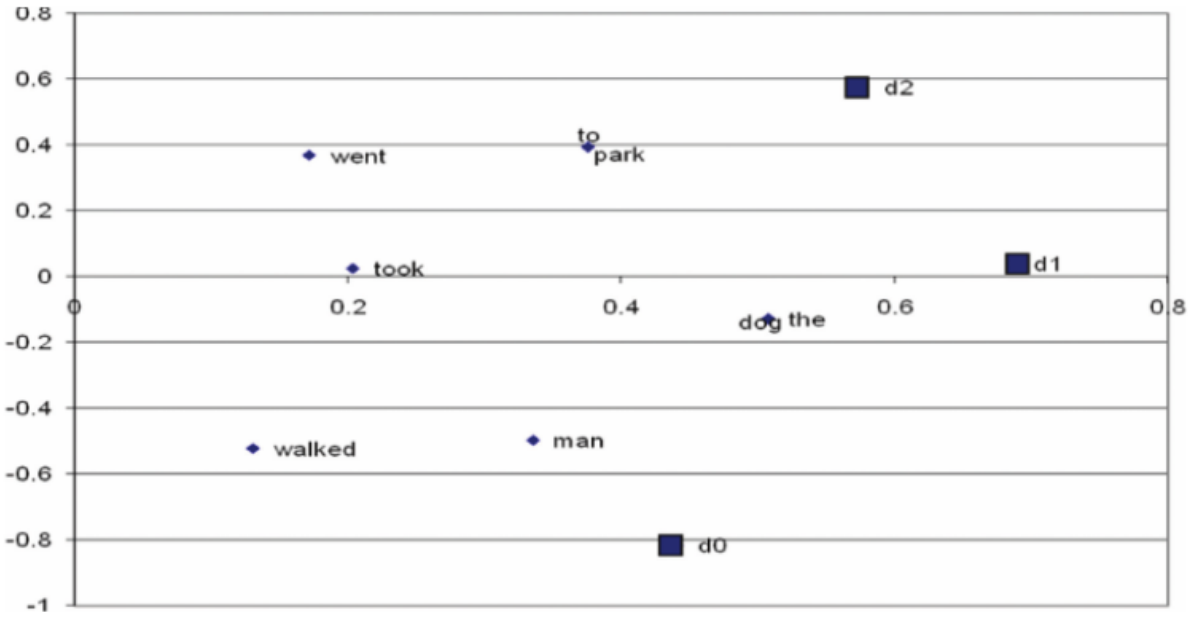

Figure 1. LSA can represent the meaning of words and sentences.

- **Root type:** the cell is filled in with the frequency of the word if its root type is a noun, otherwise the cell value is set to 0.
- **Modified Tf-idf:** this approach is proposed in Ozsoy et al., in order to eliminate noise from the input matrix. The cell values are set to tf-idf scores first, and then the words that have scores less than or equal to the average of the row are set to 0.

Step 2:

Singular Value Decomposition: SVD is an algebraic method that can model relationships among words/phrases and sentences. In this method, the given input matrix A is decomposed into three new matrices as follows:

$$A = U \sum V^{T}$$

where U = [uij] is an m × n column-orthonormal matrix whose columns are called left singular vectors; Σ = diag(σ1, σ2, …, σn) is an n × n diagonal matrix, whose diagonal elements are non-negative singular values sorted in descending order, and V = [vij] is an n × n orthonormal matrix, whose columns are called right singular vectors (see figure 2). If rank(A) = r, then Σ satisfies:

$$\sigma_1 \geq \sigma_2 \ldots \geq \sigma_r > \sigma_{r+1} = \ldots = \sigma_n = 0 .$$

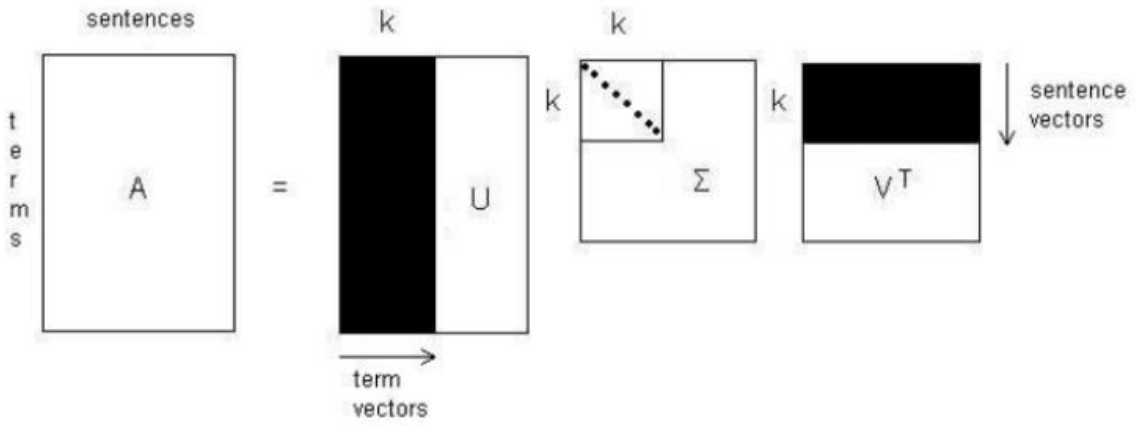

Figure 2: Singular Value Decomposition

The simplification of applying the SVD to the terms by sentences matrix A can be made from two different viewpoints. From transformation point of view, the SVD derives a mapping among the m-dimensional space spawned by the weighted term-frequency vectors and the r-dimensional singular vector space. From semantic point of view, the SVD gets the latent semantic structure from the document represented by matrix A. This operation reflects a breakdown of the original document into r linearly-independent base vectors or concepts. Each term and sentence from the document is mutually indexed by these base vectors/concepts. A unique SVD feature is that it is capable of capturing and modelling interrelationships among terms so that it can semantically cluster terms and sentences. Moreover, as demonstrated in, if a word combination pattern is salient and recurring in document, this pattern will be captured and represented by one of the singular vectors. The magnitude of the corresponding singular value indicates the importance degree of this pattern within the document. Any sentences containing this word combination pattern will be projected along this singular vector, and the sentence that best describes this pattern will have the largest index value with this vector. As each particular word combination pattern describes a certain topic/concept in the document, the facts described above obviously lead to the hypothesis that each singular vector represents a salient topic/concept of the document, and the magnitude of its corresponding singular value represents the degree of importance of the salient topic/concept. Based on the above discussion, authors proposed a summarization method which uses the matrix VT. This matrix describes an importance degree of each topic in each sentence. The summarization process chooses the most informative sentence for

each topic. It means that the k'th sentence we choose has the largest index value in k'th right singular vector in matrix VT.

Step 3:

Sentence selection: using the results of SVD, different algorithms are used to select important sentences.

LSA has several shortcomings. The first one is that it does not use the information about word order, syntactic relations, and morphologies. This kind of information can be mandatory for finding out the meaning of words and texts. The second limitation is that it uses no world knowledge, but just the information that exists in input document. The third limitation is related to the performance of the algorithm. With larger and more inhomogeneous data the performance decreases sharply. The decrease in performance is caused by SVD, which is a very complex algorithm.

Enhanced LSA Summarization:

The above described summarization method has two remarkable disadvantages. At first it is necessary to use the same number of dimensions as is the number of sentences we want to choose for a summary. However, the higher is the number of dimensions of reduced space, the less significant topic we take into a summary. This disadvantage converted into an advantage only in the case when we know how many different topics has the original document and we choose the same number of sentences into a summary. The second disadvantage is that a sentence with large index values, but not the largest (it doesn't win in any dimension), will not be chosen although its content is for the summary very suitable. In order to clear out the discussed disadvantages we suggest following modifications in the SVD-based summarization method. Again we need to compute SVD of a term by sentences matrix. We get the three matrices as shows the figure 2. For each sentence vector in matrix V (its components are multiplied by corresponding singular values) we compute its length. The reason of the multiplication is to favor the index values in the matrix V that correspond to the highest singular values (the most significant topics). Formally:

$$s_k = \sqrt{\sum_{i=1}^{n} v_{k\,i}^2 \cdot \sigma_i^2},$$

where sk is the length of the vector of k'th sentence in the modified latent vector space. It is its eminence score for summarization too. n is a number of dimensions of the new space. This value is independent on the number of summary sentences (it is a parameter of the method). In our experiments we chose the dimensions whose singular values didn't fall under the half of the highest singular value (but it is possible to set a contrary strategy). Finally, we put into a summary the sentences with the highest values in vector s.

Summary Evaluation:

Evaluation of automatic summarization in a standard and economical way is a difficult task. It is the equally important area as the own summarization process and that's why many evaluation approaches were developed.

Evaluation by Sentence Co-selection:

Co-selection measures include precision and recall of co-selected sentences. These methods recommended having at disposal a "right extract" (to which we could compute precision and recall). We can obtain this extract in several ways. The most common way is to obtain some human (manual) extracts and to declare the average of these extracts as "ideal (right) extract". However, obtaining of human extracts is usually problematic. Another problem is that two manual summaries of the same input do not usually share many identical sentences.

Content-based methods:

We can clear out the above explored weakness of co-selection measures by content-based similarity measures. These methods compute the similarity between two documents at a more fine-grained level than just sentences. The basic method is to compute the similarity among the full text document and its summary with the cosine similarity measure, computed by the following formula:

$$\cos(X, Y) = \frac{\sum x_i * y_i}{\sqrt{\sum (x_i)^2} * \sqrt{\sum (y_i)^2}},$$

where X and Y are representations based on the vector space model.

Relevance Correlation:

Relevance correlation is a measure for accessing the relative decrease in retrieval performance when indexing summaries instead of full documents.

Task-based evaluations:

Task-based evaluations measure human performance using the summaries for a certain task (after the summaries are generated). We can for example measure a suitability of using summaries instead of full texts for text categorization. This evaluation requires a classified corpus of texts.

Evaluation based on Latent Semantic Analysis:

We stratify this new method to a content-based category because, like the classical cosine content-based approach, it evaluates a summary quality via content similarity between a full text and its summary. Our method uses Singular Value Decomposition of a terms by sentences matrix, exactly matrix U. This matrix represents the degree of importance of terms in major topics/concepts. In evaluation we measure the similarity between the matrix U derived from the SVD performed on the original document and the matrix U derived from the SVD performed on the summary. For appraising this similarity we have proposed two measures.

Similarity of the Main Topic:

This method compares first left singular vectors (see figure 3) of the full text SVD (i. e. SVD performed on the original document) and the summary SVD (i. e. SVD performed on the summary). These vectors correspond to the most salience word pattern in the full text and its summary (we can call it the main topic).

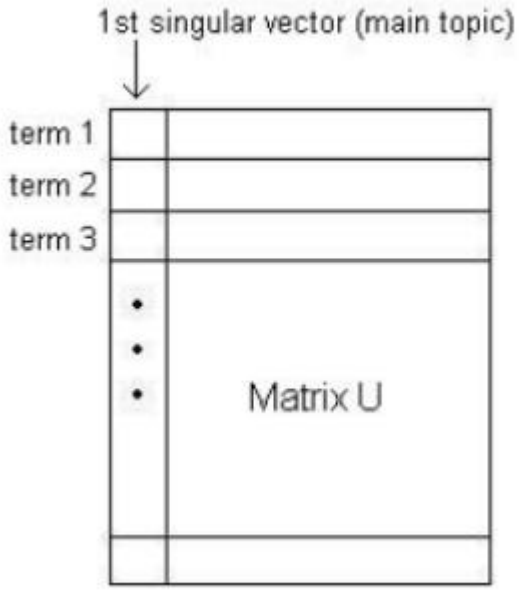

Figure 3: 1st singular vector (main topic)

Then we measure the angle between the first left singular vectors. They are normalized, so we can use the following formula:

$$\cos\varphi = \sum_{i=1}^{n} ue_i \cdot uf_i \ ,$$

where uf is the first left singular vector of the full text SVD, ue is the first left singular vector of the summary SVD (values, which correspond to particular terms, are sorted up the full text terms and instead of missing terms are zeroes), n is a number of unique terms in the full text.

Similarity of the Term Significance:

This evaluation method compares a summary with the original document from an angle of n most major topics. We propose the following process:

- Perform the SVD on a document matrix.
- For each term vector in matrix U (its components are multiplied by corresponding singular values) compute its length. The reason of the multiplication is to favor the index values in the matrix U that correspond to the highest singular values (the most significant topics). Formally:

$$s_k = \sqrt{\sum_{i=1}^{n} u_{k,i}^2 \cdot \sigma_i^2},$$

where sk is the length of the k'st term vector in the modified latent vector space, n is a number of dimensions of the new space. In our experiments we chose the dimensions whose singular values didn't fall under the half of the highest singular value (but it is possible to set a different strategy).

- From the lengths of the term vectors (sk) make a resulting term vector, whose index values hold an information about the term significance in the modified latent space (see figure 4).
- Normalize the resulting vector.

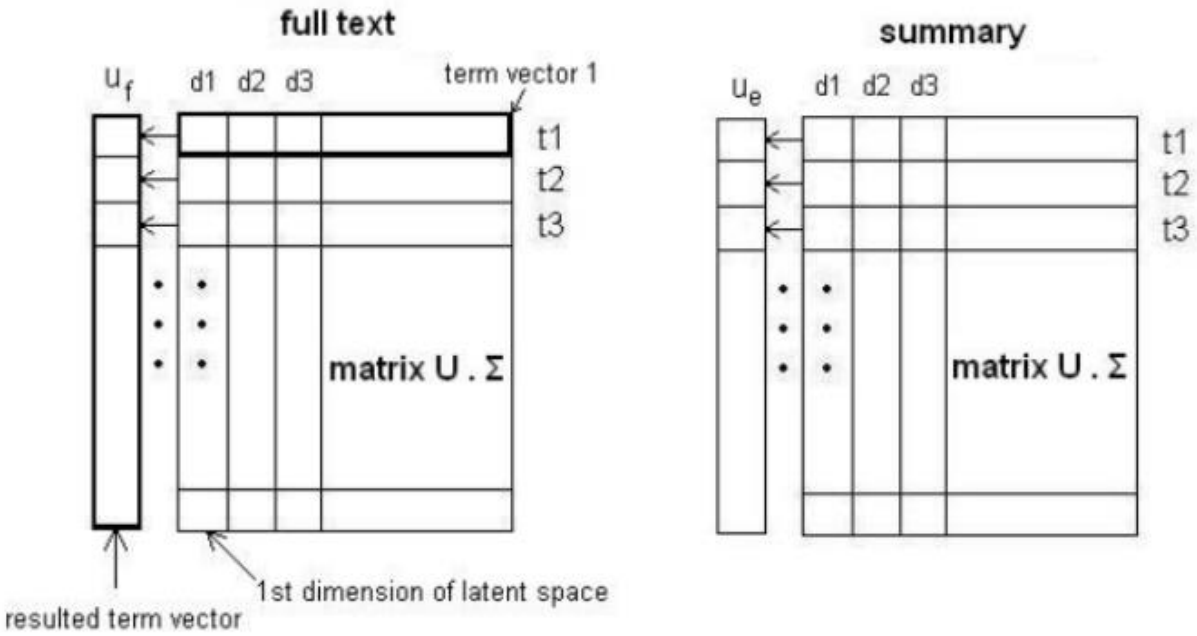

Figure 4: Creation of a resulting term vectors of a full text and a summary

This process is performed on the original document and on its summary (for the same number of dimensions according to the summary) (see figure 4). In the result, we get one vector corresponding to the term vector lengths of the full text and one of its summary. As a similarity measure we use again the angle between resulting vectors.

This evaluation method has the following advantage above the previous one. Suppose, an original document contains two topics with the quite same significance (corresponding singular values are almost the same). When the

second significant topic outweighs the first one in a summary, the main topic of the summary will not be consistent with the main topic of the original. Taking more singular vectors (than just one) into account removes this weakness.