A Survey Paper on Text Summarization Methods

Rajat K. Kulshreshtha[1], Anuranjan Srivastava[2], Mayank Bhardwaj[3]

[1,2,3]*School of Computer Science and Engineering, VIT Vellore*

Abstract - *From the last decade, it is observed that there is a never-before-seen positive flux in the quantity of textual information that a single document or even multiple documents present to us. This has created a necessity of doing extensive research in the field of Automatic Text Summarization in the field of Natural Language Processing (NLP), while this call for research work was made in 1950, the exponential growth of computing power in the 21st century has allowed unconventional methods to work better in this field. This survey tends to present a go through upon some of the most relevant approaches for summarization from surface to rhetorical, also classifying the techniques based on Single and Multiple Documents used as an input, the aim is to present a good read to the readers, young and budding linguists, about various existing methods used for text summarization.*

Key Words: Text Summarization Approaches, Hidden Markov Models, latent semantic analysis, Machine Learning Methods, Natural Language Processing

1. INTRODUCTION

The world has witnessed a tremendous rise of Internet over the last two decades, and due to this, a plethora of information that is available, is for the first time in recorded history, available on the scale comparable for a global reach, people are able to read books, they can never have access to. People are reading research papers from the authors, they never even thought of having a chance to see upon the work. But do all the people want to read all the textual content available? All the content of a book? The answer to that question is no. Most of the people don't want to read all the content, and just want to get a gist of all the content. When we, humans, summarize a textual content, what we do, first we read all the content, and then we summarize it. And this takes a lot of time, however, the research in the field is going on as far as the 1950s. One of the most notable works of that time was done by [1] the method that was proposed by them was to extract all the sentences, based on the constraints such as word and phrase frequency. The sentences of a document were extracted as a function of high-frequency common words. This simple definition of summary captures three important aspects that characterize research on automatic summarization Summaries may be produced from a single document or multiple documents, Summaries should preserve important information, Summaries should be short. Generally, there are two different approaches to summarization, namely, extraction and abstraction. While *extraction summarization* completes the process by way of *extracting* the sentences that are most probable, *abstractive summarization* on the other hand, aims to produce content in a new fresh way. In other settlement, advanced natural language techniques are used in generating abstract summaries, to define an "abstract" version of text, that defines the almost the same context as that of the main text. We humans, create the text summaries as probably a mixture of extractive and abstractive summaries, as we have the ability to retain the critical points in our memory, and at the same time we are also able to generate a gist of what we read. Apart from *Extractive and Abstractive Summarizations* summarization process is also classified on the base of *Single and Multiple Document Classification.* Single-Document Summarization has the characteristic that the flow of information in a given document is not uniform, which means that some parts are more important than others.

In basic approaches, various approaches from the oldest surface level approach to graph-based approaches, along with the corpus based, cohesion-based approaches are covered, this is followed by QR Decomposition and HMM, brief discussion about Latent Semantic Analysis is presented, and finally, Machine Learning methods are discussed.

2. BASIC APPROACHES FOR TEXT SUMMARIZATION

2.1 Surface Level Approach

The surface level approach looks at cue words and phrases like "in conclusion", "important", "in this paper" [2] or complete sentences containing them which are then rearranged to form a coherent summary. It was developed in 1958 [1].

The words are selected based on their term frequency (important sentences contain frequently appearing words), location (words and phrases in titles and headings are of relevance) and special words found in the original document.

The main advantage of these approaches is the robustness because it uses some straightforward methods to select summary sentences. However, there are some limitations in terms of the quality of the summary because it is hard to understand the real meaning of a sentence using these approaches. Also, since these methods extract the complete sentences, they cannot achieve greater compression rates compared to the deeper approaches.

2.2 Corpus-Based Approaches

The main idea of the Corpus-Based approach is that instead of looking for term frequency using the original content, a corpus is made from similar contents and relevance of a word is calculated by the formula: tf * idf, where tf is the frequency of the word in the document and idf is the inverted document frequency. [3] [4], the authors determined relevance by using WordNet [5] such that "bicycle" is accounted whenever "bicycle", "bike", "brake", "spokes" etc. are found in the text. A simple Bayesian classifier can also be used [6] to calculate the probability of a sentence from the original text being of relevance for the summary generation. The authors trained the classifier from a corpus of 188 sets of document-summary pairs in the field of science. Features like length of the sentence, position in paragraph, emphasised and capitalized words, structure of phrases, frequency of words were used for training the Bayesian classifier.

2.3 Cohesion-Based Approaches

Surface level and Corpus-based approaches fail to account relations between sentences in a document. For example, in the sentence, "I told him to make the report", the pronoun "him" could be put into the summarized text without even mentioning the person being referred to making it difficult to understand. Text cohesion contains relations among terms of the document determining text connectivity. Lexical chains are used in this method [7]. They are a grammatically independent sequence of words that express the cohesive structure of the text. An example can be: [Rome → capital → city → inhabitant].
These lexical chains provide the solution to the problem previous summarization methods faced i.e. loss of context after generation of the summary.
Lexical Chains can be formed by using WordNet databases for finding contextual relations between words and phrases [7] [8]. Scores of chains are then calculated from types and number of relations in the chain. The chains with relatively higher scores are used for generating the summary.

2.4 Rhetoric-Based Approaches

The Rhetoric based approach works on the principles of Rhetorical Structure Theory or RTS. The main idea of the text to be summarized is called the "nucleus" from which, less important text units, satellites, are connected by a rhetorical relation forming a tree. The nucleus is given a weight of "0" and all its satellites, a weight of "1" [9]. Sentences are then scored by the sum of weights from the nucleus to the last word of the sentence present as a node in the tree. During summarization, text units with high scores are extracted from different trees.
The rhetorical structure-based summarization techniques assume that the relationship between text units form a binary tree structure [10] however, a large document may have a more complicated tree structure which can make it complex to maintain and use it. This method of summarization also requires a comprehensive relation analysis among text units and intensive human interactions [10] [9] which again extend the difficulty in its use as compared to other methods.

2.5 Graph – Based Approach

In this method, each sentence of the given text is represented as a vertex and edges between these vertices are formed using sentence similarity relation, where similarity is measured from the content overlap of two sentences which is simply the number of common tokens between lexical representations of those two sentences. After the ranking algorithm is run on the graph recursively, the sentences are sorted in the reverse order of their score, and the top sentences are then used to form the summary [11].
HITS algorithm [12] and Google's PageRank algorithm [13] are a prime example of this method. They have been successfully used for analysis of the link structure of the Web, citation analysis and in various social networks.

3. SINGLE DOCUMENT SUMMARIZATION

3.1 Naive-Bayes Methods

[6] Has told a method, which trains with data, that is it shows a clear conscience of machine learning being involved. The classifier categorizes each sentence to be deemed for putting in the summary or not. Let s be a particular sentence, S the set of sentences that make up the summary, and $F_1 \ldots\ldots F_k$ the features, now assuming the independence of the features:

$$P(s \in S \mid F_1, F_2, \ldots F_k) = \frac{\Pi_{i=1}^{k} P(F_i \mid s \in S) \cdot P(s \in S)}{\Pi_{i=1}^{k} P(F_i)}$$

Using the above equation each sentence is given a score, and only the top n desired scoring sentences were extracted. To train the system, manual abstracts of documents in the following fashion: In the manual abstract the match was manually analysed. The system extracts were then evaluated against the mapping created by authors. [14] also used a naïve bayes classifier, it was used in their DimSum system, which used the factors such as term frequency (tf) and inverse document frequency (idf) to derive words of importance.

3.2 Log-Linear Models

[15]'s claim came after [16]'s Markov Model based extraction which is discussed later in this paper. The claim was that uptil the 2000s, all the models assumed feature independence, in the paper, the authors have used log linear models which is as follows: Let 'c' be a label, 's' the item we are interested in labelling, fi the i-th feature, and i the corresponding feature weight.

$$P(c \mid s) = \frac{1}{Z(s)} exp\left(\sum_i \lambda_i f_i(c,s)\right)$$

where, $Z(s) = \sum_c exp(\sum_i \lambda_i \ f_i \ (c,s))$

.

There are only two possible situations in this case either the sentence is extracted or not, which is also quite similar to the HMM Model presented by [16], as it also presents this binary condition. The weights were trained by conjugate gradient descent.

3.3 Deep Natural Language Analysis Methods

[7] Formulated a work that require considerable amount of linguistic analysis. To understand their method in a better way, lexical chains were used: lexical chains are the sequences of the relatable words in the text, which can either be short as adjacent words, or as long as the entire document. The following steps are deployed segmentation of the text, identification of lexical chains, and using strong lexical chains to identify the sentences worthy of extraction. To find out lexical chains, the authors used Wordnet, applying three generic steps: Selecting a set of candidate words. For each candidate word, finding an appropriate chain relying on a relatedness criterion among members of the chains. If it is found, inserting the word in the chain and updating it accordingly.

4. MULTI-DOCUMENT SUMMARIZATION

Multi-document summarization, as the name suggests, follows the approach of extracting the summary from multiple documents. It became popular in 1990s for summarizing news articles. Today, several online news summarization techniques are derived from it and Google news is a prime example of that.

4.1 Abstraction and Information Fusion

The first proper implementation of multi-document summarization technique was done with SUMMONS. It was used in the area of interest of terrorism and terrorist related news articles and it produced a summarized report consisting of relevant information of each terror event being merged. The reports were also collected from different news agencies.

SUMMONS had two layers of work namely, a content planner and a linguistic generator. The content planner uses multiple templates as input from which, it selects the information relevant for the summary. The linguistic generator chooses the perfect combination of words, which forms grammatically correct phrases and sentences, and forms a coherent text.
This was improved later by and [7], where the input is now a set of related documents in raw text, like those retrieved by a standard search engine when a query is asked.

4.2 Graph Spreading Activation

[17] Formulated a new methodology for summarization based on a graph approach for information selection. In this approach, every word in the document is converted into nodes representing their singular occurrence (i.e., one word together with its position in the text). These nodes further have relational links like “adjacency” links (linking adjacent word), “same” links (linking references of the same word), “alpha” links (encodes semantic relationships retrieved from WordNet and NetOwl118), “phrase” links (linking sequences of adjacent nodes belonging to the same phrase), and “name” and “coref” links for linking co-referential name occurrences
Once the graph is built, “topic” nodes are identified by using stem comparison and then they become the entry nodes.

5. QR DECOMPOSITION, HIDDEN MARKOV MODELS FOR TEXT SUMMARIZATION

Information based Retrieval systems have been doing rounds from late 1995, and thus have been gaining wider attention [18]. Microsoft word has also a generic summarizer. This is an example of a generic summary, one that shows us what are the must-have points or take away from the document. Web pages show a summary of the documents (web pages) to give us a rough idea, on what the page will be looking like. These query-based summaries are more informative, as they directly are the product of user entered query.

Indicative and Informative are the two types of summaries that are discussed. Indicative summaries are one or two-line-based summaries, that gives the reader a sense that he/she should read the document or not. Informative summaries are the summaries which may or may not be on par of the size of the document, there is no length barrier, but they contain very concise information about the document [18].

A very vast work using HMM in text summarization comes from [16], the authors summarize the document by seeking the main ideas. Going beyond the words, and instead focuses on terms [19]. Disambiguation is removed from the words by Co-location. After that, a term sentence matrix is generated, here sentences are viewed as vectors, and then, the job of the automatic summarization system is to choose a subset which is relatively small, from the pool of the vectors, and selects only those who conform with the essence of our main idea. This very idea is heavily focused on the mathematical concept of QR Vectorization in Linear Algebra.

In the first algorithm that is carried out using QR with partial pivoting, the focus is on how to define an idea, in this paper, the authors have used term as an idea. An Idea is important in that sentence if it appears in the sentence. Otherwise, the importance is zero. The term

sentence matrix in which the non-zero values are taken as ones. In term sentence matrix each column represents a sentence. The entry A_{ij} of the matrix represents that i^{th} term in j^{th} sentence is present than it is one, else it is zero.

For choosing to sentence Euclidean Length of a sentence with a term, is used as a measure of importance. Therefore, the sentences that have many terms, are related as important. At each iteration, the sentence that is chosen is the one which is there with largest norm, or Euclidean measure. This choice is called Pivot. Complete QR decomposition is not utilized in the process, we decompose or add up to the k^{th} sentence, which is the required length of our summary. And that becomes the stopping criteria for the decomposition [16].

The next algorithm that was shown in the paper uses Hidden Markov Models. The basic idea behind this algorithm is to compute an a – posterior probability, that each sentence is the part of the summary, each sentence is a "probable" contender to be contained in the summary. The HMM has limited assumptions on independence, as can be seen in the case of Naïve Bayes Classifier [6]. Particularly, HMM doesn't obey the condition of Naïve Bayes Classifier that the probability of inclusion of i^{th} sentence doesn't depend on the probability of inclusion of $(i-1)^{th}$ sentence. A picture of a markov chain is given in Figure 1.

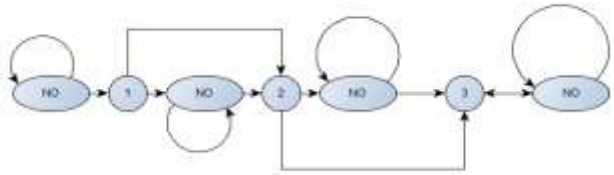

Figure 1 Summary Extraction Model which extracts two lead sentences.

The HMM for text summarization was made considering five features. The positioning of a sentence in a document. The position of a sentence within its containing paragraph, 1 for the first sentence, 3 for the last sentence, and 2 for an intermediate solution. Number of terms in a sentence. The last two features are more of a TF-IDF matrix, called baseline probability, and Document term probability.

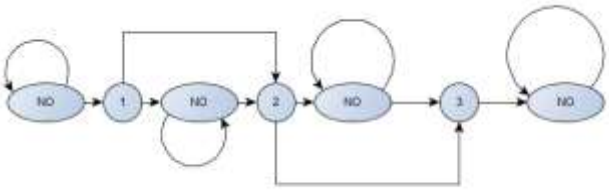

Figure 2 Summary Extraction Markov Model to extract three sentences.

The chain in Figure 2 differs from Figure 1, as can be seen. The Markov Model that is built upon the features that are described previously, one possible, but less suited way, is to use Naïve Bayes Classifier. But that poses a threat against the idea presented by the author, the idea involves dependencies, that too in calculating probabilities on the inclusion of the next sentence based on whether the current sentence. Naïve Bayes' primitive assumption is Independence, which contradicts with the idea of the HMM. Both the algorithms that summarize, agree with each other at 75%. Which is a very high agreement percentage compared to the human agreement [16].

6. LATENT SEMANTIC ANALYSIS BASED SUMMARIZATION

Latent Semantic Analysis, in layman's term can be explained as finding order from chaos. It is an unsupervised method for excavating the meaning of text, i.e. semantics, on observed words. It keeps information about which words are used in sentence and reserve information of common word amongst sentences. Here order is the similarity ranking of the sentences.

The key point of LSA is it avoids the problem of synonyms. Using LSA all the main topics of a documents are covered. A method was proposed by [20] for using LSA in news articles categorization. In this method first, we build a term sentence matrix, it is n by m matrix, where n stands for number of input words, and m stands for m sentences. An entry A_{ij} in the matrix is weight of word i in the sentence j. It is computed with TF–IDF technique. Then SVD is applied on the matrix and transforms *A* as:

$$A = U\Sigma V^T$$

where *U* is term topic matrix and is *m by r*, then Σ is a diagonal *r by r* matrix consisting of all the Eigenvectors, and V^T is a matrix that is topic-term matrix *r by n*. A pictorial representation of above equation is given in Figure 3. The basic intuition about Latent Semantic Analysis is that, it transforms the given document, which is mostly a work of human language and is asymmetrically distributed, into an ordered matrix representation by Eigen Vectors by exploiting its hidden semantic network. Apart from Text Summarization, It is also used in indexing.

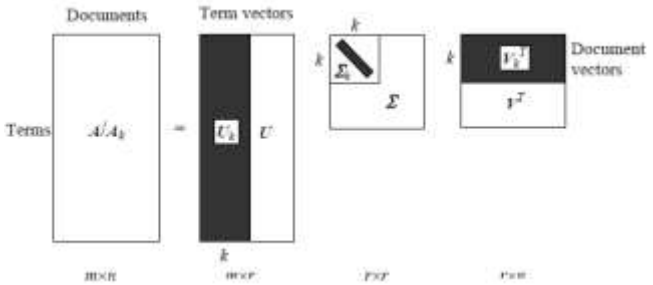

Figure 3 Pictorial representation of Latent Semantic Indexing [21].

The weight of each topic to determine the relative proportion of the summary that should cover the topic, thus allowing for a variable number of sentences per topic. Another improvement was to notice that often sentences that discuss several of the important topics are good candidates for summaries. While Gong and Liu's Method was the main study in LSA based extraction, the approach of [22] starts firstly the input matrix is created and then the SVD Calculation is performed. The step that differs from Gong and Liu is the sentence selection step, which comes next. Both V and matrices for sentence selection are used in this approach. In this approach, the length of each sentence vector, represented by the row of V matrix, is used for sentence selection.

$$g(s_i) = \sqrt{\sum_{j=1}^{m} d_{ij}^2}$$

The length of i^{th} sentence is calculated on the indexes which are lower. Matrix is used as a multiplication parameter to give more emphasis to the most important concepts. The sentence with the highest length value is chosen to be a part of the resulting summary.

7. CONCLUSION

The expanding development of the Internet has made an immense measure of data accessible. It is difficult for people to outline a lot of content. Along these lines, there is a huge requirement for text summarization in this time of oceans of information. In this paper, we emphasized different text summarization methodologies. We portrayed the absolute most broadly utilized strategies, for example, subject portrayal approaches, recurrence driven strategies, chart based and machine learning procedures. Although it isn't doable to clarify every differing calculation and methodologies completely in this paper, we think it gives a decent knowledge into late patterns and advances in programmed rundown strategies and depicts the cutting edge in this exploration territory.

REFERENCES

[1] H. P. Luhn, "The automatic creation of literature abstracts," IBM Journal of research and development, vol. 2, pp. 159-165, 1958.

[2] P. B. Baxendale, "Machine-made index for technical literature_an experiment," IBM Journal of Research and Development, vol. 2, pp. 354-361, 1958.

[3] C.-Y. Lin and E. Hovy, "Automatic evaluation of summaries using n-gram co-occurrence statistics," in Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1, 2003.

[4] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," Text Summarization Branches Out, 2004.

[5] G. A. Miller, "WordNet: a lexical database for English," Communications of the ACM, vol. 38, pp. 39-41, 1995.

[6] J. Kupiec, J. Pedersen and F. Chen, "A trainable document summarizer," in Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval, 1995.

[7] R. Barzilay and M. Elhadad, "Using lexical chains for text summarization," Advances in automatic text summarization, pp. 10-17, 1999.

[8] R. Barzilay and N. Elhadad, "Inferring strategies for sentence ordering in multidocument news summarization," Journal of Artificial Intelligence Research, vol. 17, pp. 35-55, 2002.

[9] K. Ono, K. Sumita and S. Miike, "Abstract generation based on rhetorical structure extraction," in Proceedings of the 15th conference on Computational linguistics-Volume 1, 1994.

[10] D. Marcu, "From discourse structures to text summaries," Intelligent Scalable Text Summarization, 1997.

[11] R. Mihalcea and P. Tarau, "A language independent algorithm for single and multiple document summarization," in Companion Volume to the Proceedings of Conference including Posters/Demos and tutorial abstracts, 2005.

[12] J. M. Kleinberg, "Hubs, authorities, and communities," ACM computing surveys (CSUR), vol. 31, p. 5, 1999.

[13] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," Computer networks and ISDN systems, vol. 30, pp. 107-117, 1998.

[14] B. Larsen, "A trainable summarizer with knowledge acquired from robust NLP techniques," Advances in automatic text summarization, vol. 71, 1999.

[15] M. Osborne, "Using maximum entropy for sentence extraction," in Proceedings of the ACL-02 Workshop on Automatic Summarization-Volume 4, 2002.

[16] J. M. Conroy and D. P. O'leary, "Text summarization via hidden markov models," in Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, 2001.

[17] I. Mani and E. Bloedorn, "Multi-document summarization by graph search and matching," arXiv preprint cmp-lg/9712004, 1997.

[18] U. Hahn and I. Mani, "The challenges of automatic summarization," Computer, vol. 33, pp. 29-36, 2000.

[19] C. Aone, M. E. Okurowski, J. Gorlinsky and B. Larsen, "A scalable summarization system using robust

NLP," Intelligent Scalable Text Summarization, 1997.

[20] Y. Gong and X. Liu, "Generic text summarization using relevance measure and latent semantic analysis," in Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, 2001.

[21] B. Liu, Web data mining: exploring hyperlinks, contents, and usage data, Springer Science & Business Media, 2007.

[22] J. Steinberger and K. Jezek, "Sentence Compression for the LSA-based Summarizer," in Proceedings of the 7th International conference on information systems implementation and modelling, 2006.