Extractive Text Summarization Techniques

Sarthak Parakh[1], Shivam Goyan[1], Somya Jain[1], Kavita Namdev[2]

[1]B.Tech. student, Dept. of Computer Science and Engineering, Acropolis Institute of Technology & Research, Indore, (M.P), India
[2]Senior Assistant Professor, Dept. of Computer Science and Engineering, Acropolis Institute of Technology & Research, Indore, (M.P), India

---***---

Abstract – *Text summarization is a process of retrieving crucial information in a concise and precise manner from original and voluminous texts while maintaining the overall meaning of the text. Data is growing exponentially day by day and so the textual data, which may be structured or unstructured, and the best way to use them is by skimming the results. We can access an immense amount of information, however, most of it is redundant, trivial, and may not deliver intended results. Using text summarization techniques can amplify the readability of text documents, reduce the investment of time in scrutinizing the information, and can increase the amount of information to be inserted in a particular domain.*

Key Words: Extractive Text Summarization, Computationally, Inverse Document Frequency, Tokenization, Stemming, Euclidean Space, Defuzzification.

1. INTRODUCTION

A summary is a condensed version of the original text, which conveys vital information in short, while preserving its key meaning. Since manual text summarization is a tedious task that can be biased, the automation of text summarization is gaining traction and tends to be a bold reason for academic research.

Automatic text summarization is a process of minimizing a band of data computationally, to generate a subset that carries the crucial and significant information from the original text with its essential meaning. Moreover, images and videos can also be summarized. As text summarization finds the most relevant sentences from the document, image summarization finds the most relevant image from the image pool, video summarization extracts the crucial frames from the video content. The most important advantage of using a text summarizer is that it increases readability and reduces the time investment. In general, automatic text summarizers select important sentences from the document and organize them together. The goal is to generate a shorter version with the same overall meaning of the document. Automatic text summarization is prevalent in the field of Natural Language Processing (NLP).

Text summarization can be comprehensively categorized into two categories: extractive summarization and abstractive summarization. Extractive summarization takes important sentences directly from the document without any alteration from the original document and groups them together. Abstractive summarization can generate new shorter text, which may or may not be part of the original document, which is rephrased, that presents the most important information from the document [1][2][3].

1.1 Extractive Text Summarization

The Extractive approach of summarizing text data involves choosing up the most important sentences and phrases from the documents. All the important phrases are combined to form a summary. So, in this case, every word and sentence in the summary is chosen from the original document without changing the context and meaning of the same [1].

2. Techniques for Extractive Text Summarization

Text summarizers extract the key sentences from the source text and concatenate them to form a concise summary. There are various automation techniques for Extractive Text Summarization which preprocesses origin data to extract the most relevant sentences and phrases out of it to include in the summary. Following are some techniques:

2.1 Term Frequency - Inverse Document Frequency

TF-IDF is a short form of Term Frequency-Inverse Document Frequency, it is a statistical method that estimates whether the word is important or not in the document in a collection of documents or corpus. In general TF-IDF value increases comparatively whenever we find that word in the document but goes down if the word frequency increase in the corpus. TF-IDF is calculated by multiplying two metrics i.e., TF value and IDF value [4][13].

$$TF(x) = \frac{\text{(Number of times term x appears in a document)}}{\text{(Total number of terms in the document)}}$$

$$IDF(w) = \frac{\log_e \text{(Total number of documents)}}{\text{(Number of documents with term x in it)}}$$

$$TF\text{-}IDF(w) = TF(w) * IDF(w)$$

2.2 Cluster Based Method

Documents are generally written to address different subject themes in a particular order. Therefore, the summary should also address different themes mentioned in the document.

So, in that case it becomes necessary to cluster out the different subject themes. Sentence preference criteria should be subjected to the similarity between the sentences of the cluster (C_x), another factor includes sentence preference according to the sentence location in the input document (L_x), and the last factor is the sentence score which gives weightage to the sentence with the similarity to the first sentence in the input document to which it belongs (F_x). The overall score (S_x) of a sentence 'x' is a weighted sum of the above three factors [3][5]:

$$S_x=W_1*C_x + W_2*F_x + W_3*L_x$$

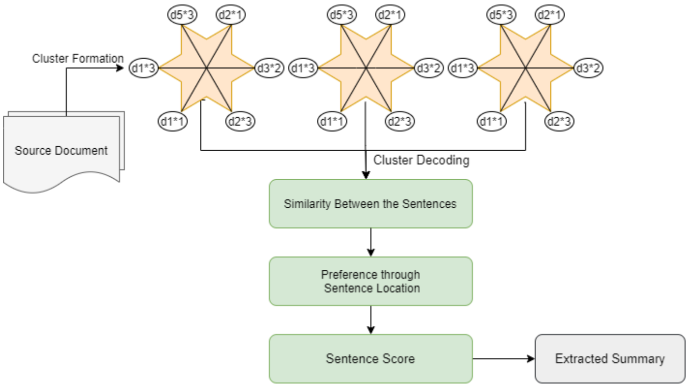

Fig -1: Flow-Chart of Cluster Based Algorithm

2.3 Graph-Theoretic Approach

The Graph Theoretic Approach tokenizes sentences from the document and words from the tokenized sentences to pre-process the entire document which then is tagged with the parts of speech to remove stop words and noisy sentences.

Features can be extracted from the represented graph like distance between the nouns:

$$\text{distance } (n_1, n_2) = | \text{position}(n_1) - \text{position}(n_2) |$$

$$\text{weights of the edges } e\ (n_1, n_2) = 1/\ (1+(\text{distance } (n_1, n_2))$$

where n_1, n_2 are nouns.

These features can help in finding the relevance score of each noun by summation of weights of all the edges (ΣN_i) and the sentence score is being calculated by summing the relevance score [3][6]:

$$\text{Sentence Score}(s) = \forall n \in s\ \Sigma\ \text{relevance}(n).$$

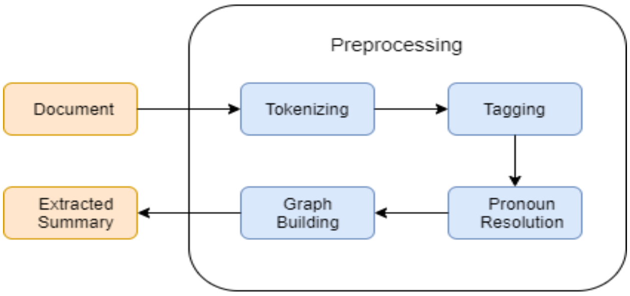

Fig -2: Flow Diagram of Graph–Theoretic Approach

2.4 Machine Learning Approach

This summarization process is a classification model which models document into two categories summary and non-summary sentences. This is done through two well-known machine learning algorithms, namely Naive Bayes and C4.5. These algorithms are passed to summary sentences which are provided after preprocessing of the document sentences. Preprocessing involves calculating TF-IDF for each word in a sentence, normalized sentence length containing main keywords gained through TF-IDF and title keyword extraction, sentence-to-sentence cohesion, and removing redundant data through stemming, case-folding, and stop-word removal.

After preprocessing summary sentences are labeled as 'positive' and the remaining sentence as 'negative'. Positive sentences are converted to their vector representation and machine learning trainable algorithm is applied.

$$P\ (s \in <S \mid F1, F2..., FN) = \frac{P\ (F1, F2, ..., FN \mid s \in S) * P\ (s \in S)}{P\ (F1, F2, ..., FN)}$$

Here, (F1, F2..., FN) are vector functions, s = training sentence, S=Summary Sentence.

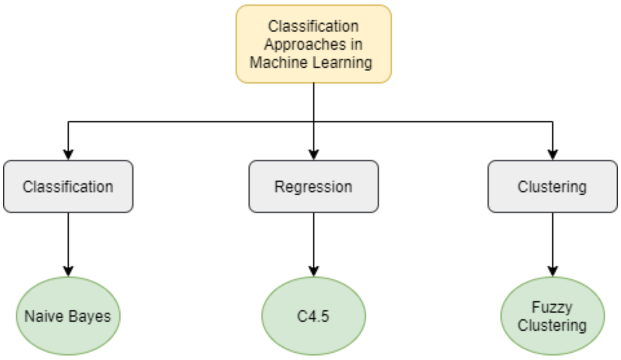

Fig -3: Machine Learning Classification Model

This statistically analyzes the classification probabilities of the sentences.

Sentences with high probabilities are added to the summarized output [3][7].

2.5 Latent Semantic Analysis

The Latent Semantic Analysis Algorithm analyzes the document by creating its word matrices and cluster the data semantically through SVD (Singular Value Decomposition) to get the most common words and sentences.

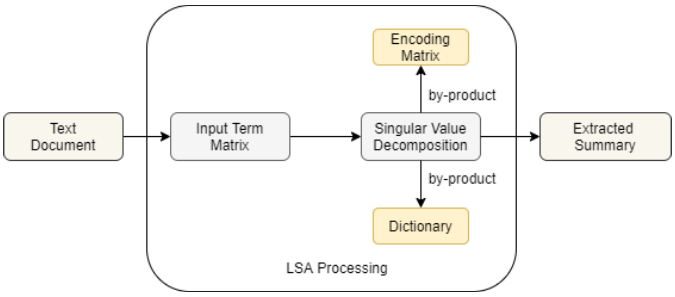

Fig -4: System Architecture – Latent Semantic Analysis

This algorithm consists of three major steps:

1. Input Matrix Creation: The document is converted into a term matrix with a cell representation method. Different approaches pertain to fill cell values like word frequency, binary representation (0/1) (representing (1) as existing word and (0) to non-existing word), TF-IDF value, and Log entropy value for each cell.

2. Singular Value Decomposition (SVD): It is an algebraic method that models the relationship between the words and the sentences. In this model term matrix values are represented as points in Euclidean space known as vectors.

3. Summary Output: In this step, the summary is extracted by selecting important sentences through the results of SVD calculations and minimized term matrix [8].

2.6 Text Summarization with Neural Networks

This method is also depicted as a classification problem that classifies whether a sentence should be included in the summary or not. It compares each sentence in the document with every other sentence and depicts a threshold score for each sentence on this basis.

It uses a feed-forward neural network consisting of three layers: Input Layer, Hidden Layer, and Output Layer. The document is fed into the input layer in sentence-wise order, all the computation work is done in the hidden layer and the output layer provides probability vectors for the sentences. Higher the probability, the higher the chances of the sentence to be categorized under the summary. Run is conducted for each page in iterative order and a fixed number of sentences are selected in each run according to the size of the summary.

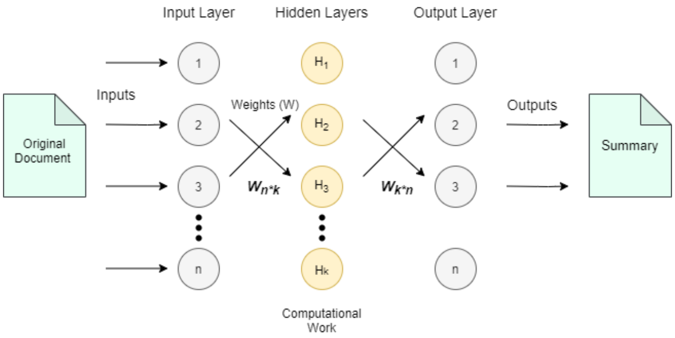

Fig -5: 3-Layer Feed Forward Neural Network

This method proves very useful for extractive summarization and could be run with limited computational power [9][14].

2.7 Fuzzy Logic

This method includes different characteristics as a measure to determine the important sentences such as similarity to the title, sentence length, term frequency, etc. as an input in the fuzzy system.

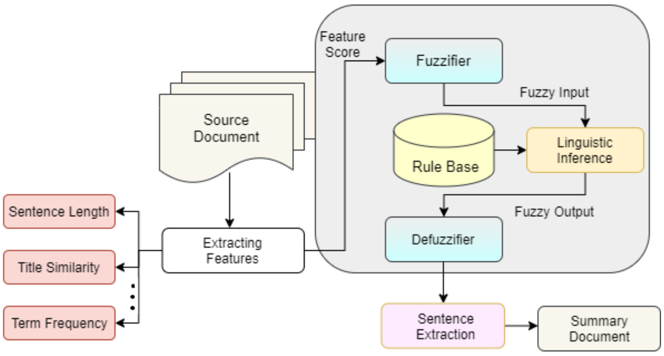

Fig -6: System Architecture of Extractive Summarization using Fuzzy Logic Approach

Using these characteristics, the scores are generated for every sentence using the fuzzy logic method. This method uses fuzzy rules and a triangular membership function. Triangular membership function labels sentences into three values, low, medium, or high and fuzzy rules determine whether a sentence is important, average, or unimportant. This is termed as defuzzification.

Then, all the sentences are ranked and the top 'n' sentences with the highest scores are extracted to be included in the summary document. Finally, the summary is arranged according to the content organized and conveyed in the original document without changing its matter of context [10][11].

2.8 Query Based Approach

Query-based text summarization includes text preprocessing initially, using the MDL (Minimum Description Length) Principle which compresses data according to the regularity in the set of data models using query-dependent minimization of description's bit length L(MQ) + L(D|MQ); where M is model, Q is query and D is a database which is encoded by the model.

Data setup for query processing is done by processing data and query through sentence splitting, tokenization, stemming, and stop-words removal. Data encoding is done in the coding table (CT) from the database D (D, CT) = L(CT) + L(D|CT). The most appropriate data set D|CT is provided sentence rankings and the sentences with the highest-ranking are being provided in the Summary [12].

3. CONCLUSIONS

Extractive Text Summarization through automation approaches depends upon the semantic analysis of the data in the document. So, in this way, the document sentences are gathered, and parameters are checked for word sense, keywords, similarities to the titles, words, and sentence occurrences, to sense utilities of covering sentences.

The fusion of various techniques will provide the high potential results as a summarized output. With the ever-growing data in the environment, reduction of redundant and un-relevant data is very crucial, otherwise handling these data will become very difficult and will lead to loss of important information.

REFERENCES

[1] D. K. Gaikwad, C. N. Mahender, "A Review Paper on Text Summarization", International Journal of Advanced Research in Computer and Communication Engineering, Vol. 05, Issue 03, Mar 2016.

[2] P. Devihosur, Naseer R, "Automatic Text Summarization Using Natural Language Processing", International Research Journal of Engineering and Technology, Vol. 04, Issue 08, Aug 2017.

[3] Saranyamol CS, Sindhu L, "A Survey on Automatic Text Summarization", International Journal of Computer Science and Information Technologies, Vol. 5 (6), 2014.

[4] K. D. Patil, S. A. Patil, Y. S. Deshmukh, "An Extractive Approach for English Text Summarization", International Journal of Sciences & Applied Research, IJSAR, 6(5), 2019

[5] R. Barzilay, K. R. McKeown, M. Elhadad, "Information Fusion in the context of Multi-document Summarization", Association for Computational Linguistics, Jun 1999, retrieved from: https://www.aclweb.org/anthology/P99-1071.pdf

[6] A. A. Natesh, S. T. Balekuttira, A. P. Patil, "Graph Based Approach for Automatic Text Summarization" International Journal of Advanced Research in Computer & Communication Engineering, Vol. 05, Special Issue 02, Oct 2016.

[7] J. L. Neto, A. A. Freitas, C. A. A. Kaedtner, "Automatic Text Summarization Using Machine Learning Approach", 2002. SBIA '02: Proceedings of the 16th Brazilian Symposium on Artificial Intelligence: Advances in Artificial Intelligence.

[8] M. G. Ozsoy, F. N. Alpaslan, I. Cicekli , "Text Summarization using Latent Semantic Analysis", Journal of Information Science, Jun 2011

[9] Sarda A.T, Kulkarni A.R, "Text Summarization using Neural Networks and Rhetorical Structure Theory", International Journal of Advanced Research in Computer and Communication Engineering, Vol. 04, Issue 06, Jun 2015.

[10] R. S. Dixit, Prof. Dr. S.S. Apte, "Improvement of Text Summarization using Fuzzy Logic Based Method" IOSR Journal of Computer Engineering, Vol. 05, Issue 06 Sep-Oct 2012.

[11] P. D. Patil, Prof. N. J. Kulkarni, "Text Summarization using Fuzzy Logic", International Journal of Innovative Research in Advanced Engineering (IJIRAE), Vol. 01, Issue 03, May 2014 (Special Issue).

[12] N. Vanetic, M. Litvak, "Query Based Summarization using MDL Principle", Association for Computational Linguistics ISBN 978-1-945626-41-8, retrieved from: https://www.aclweb.org/anthology/W17-1004.pdf

[13] A. Jain, "Automatic Extractive Text Summarization using TF-IDF", Web Blog Post - medium.com, Apr 2019.

[14] S. Chaudhary, "Extractive Text Summarization Using Neural Networks", Web Blog Post - heartbeat.fritz.ai, May 2018.

[15] https://app.diagrams.net/

BIOGRAPHIES

Sarthak Parakh (B.Tech.) student Dept. of Computer Science & Engineering, Acropolis Institute of Technology & Research, Indore (M.P), India

Shivam Goyan (B.Tech.) student Dept. of Computer Science & Engineering, Acropolis Institute of Technology & Research, Indore (M.P), India

Somya Jain (B.Tech.) student Dept. of Computer Science & Engineering, Acropolis Institute of Technology & Research, Indore (M.P), India

Kavita Namdev, Sr. Asst. Prof. Dept. of Computer Science & Engineering, Acropolis Institute of Technology & Research, Indore (M.P), India