A Novel Approach To Text Summarization Using Machine Learning

.

ABSTRACT

Text summarization is a key strategy in the domains of information retrieval and natural language processing (NLP). Its objective is to reduce a lengthy written document into a clearer, more succinct summary of the information it contains. When a text document is too lengthy or intricate to analyse completely, as in news stories, academic papers, or legal documents, this approach is extremely helpful. The major challenge of text summarising is to take the most important and relevant information from the original text and convey it in an understandable and concise way. In this study, extractive and abstractive summarising techniques are the two primary categories of text summary methods. The paper also presents several algorithms that have been proposed for text summarization, including TextRank, Seq2Seq, and BART. TextRank is a simple and fast algorithm that works well for short documents, Seq2Seq is a deep learning-based approach that generates high-quality summaries, and BART is a transformer-based algorithm that provides the best results on benchmark datasets. The obtained ROUGE Score after passing TextRank, BART, and Seq2SEq algorithm significant also..

Keywords: Text Summarization, Machine Learning, BART,TextRank, Text analysis, NLP

1. INTRODUCTION

Natural language processing (NLP), a discipline of artificial intelligence and computer science, is the study of how computers and people communicate using natural language. The objective of NLP is to create models and algorithms that can process and comprehend human language, making it possible for computers to carry out activities like sentiment analysis, text categorization, machine translation, and text summarization, among others.

NLP is a complex and challenging field, as human language is rich, ambiguous, and constantly evolving. To address these challenges, NLP relies on a variety of techniques, including statistical models, machine learning, deep learning, and rule-based systems. These techniques are used to process and analyze text data, extract meaningful information, and generate natural language outputs.

You can reduce a lengthy written document into a shorter, more manageable representation of its content by using text summarization. There are many different approaches to text summarising, such as extractive and abstractive techniques [24] [25]. The summary is created using extractive summarization, which pulls out the key phrases or clauses from the original text. Contrarily, abstractive summarization starts from scratch with a new summary depending on the content of the original text. The best strategy to use relies on the particular needs of the summarising work because both techniques have advantages and disadvantages [26].

Text summarization has numerous applications, including information retrieval, content-based recommendation systems, and knowledge management. In the field of information retrieval, text summarization can be used to provide a quick overview of search results, allowing users to quickly identify relevant documents. In content-based recommendation systems, text summarization can be used to provide users with a brief description of the content of recommended documents. Finally, in knowledge management, text summarization can be used to extract key information from large and complex documents, making it easier to store, manage, and retrieve information [27].

Text summarising is a difficult topic since it calls for the capacity to extract the most crucial information from the source text and to display it in a clear and simple manner.[16] A good summarization should accurately capture the main ideas and information contained in the original text, while excluding irrelevant or redundant information. To do this, text summarising algorithms must have a thorough grasp of the architecture and content of the original text and be able to recognise the most crucial clauses or phrases based on their significance, applicability, and coherence [28].

For text summarization, a number of methods have been put forth, including TextRank, Seq2Seq, and BART. The text document's structure is used by the graph-based algorithm TextRank to obtain a summary. It models the text as a graph of words and sentences, and calculates the importance of each node based on the number and weight of its edges. Seq2Seq is a deep learning-based algorithm that uses the encoder-decoder architecture to generate a summary. The transformer-based algorithm BART (Bidirectional Encoder Representations from Transformers) processes the input text in a bidirectional manner.

2. LITERATURE REVIEW

This section cites earlier works that make use of the various summarizing methods. Instead of sentence production for text summary, the majority of researches focus on sentence extraction. The most popular approach of summarization creates extractive summaries based on statistical aspects of the sentence.

According to Luhn[4], the words that are used the most often in a text correspond to its most crucial ideas. He wanted to assess each phrase based on the frequency of each word before selecting the best result. Methods based on location, title, and cue words were suggested by Edmunson[16]. He argued that the summary should include the topic information, which is usually found in the opening few words or paragraphs of a text. One flaw in the statistical technique is that it ignores the semantic relationships between sentences. In order to give a summary, Goldstein [2] developed a query-based summarizing technique that would extract important lines from a text according to the query fired. There is a suggested query for the extraction criterion. The more words combined in the question and a sentence, the more likely it is to be included in a summary. Goldstein[2][1] used statistical and linguistic characteristics to analyze the summaries of news stories in order to assess the phrases in the document. One approach to summarizing is sentence extraction and grouping. In order to determine how similar sentences' cumulative phrases are, sentences should first be clustered depending on how far apart they are from one another semantically, according to ZHANG Pei-ying and LI Cun[5]. Finally, the sentences should be selected using extraction procedures. K-means algorithm is used to group the sentences together[5]. Morris and Hirst[9][7] were the authors who initially developed the idea of lexical chains. Lexical chains [7] take advantage of the relationships between any number of related words. By assembling groups of semantically similar words, we can form lexical chains. Barzilay and Elhadad[8][6] built a lexical chain by utilising WordNet to determine the semantic distance between terms. The phrases associated with the chosen strong lexical chains are picked as a summary.

Using a liner time method, H. Gregory Silber and McCoy [10] created a method for creating lexical chains. By creating an intermediate representation, the author follows Barzilay and Elhadad's [6] approach of using lexical chains to extract crucial concepts from the original text. The method for using lexical chaining to construct an array of Meta-Chains whose size is equal to the number of noun senses in the Word Net and the document is detailed in the article [10]. Proper nouns and anaphora resolution issues with the algorithm needs to be fixed. A alternative approach to summarization is found in graph theory [11]. To create a semantic network of the original text, the author suggested a technique based on subject-object-predicate (SOP) triples from individual phrases. Every word has essential, significant concepts strewn throughout it. According to the author [11], by identifying and using the links between them, it may be feasible to recover crucial information. Pushpak Bhattacharyya [12] of the IIT Bombay, one of the researchers, proposed a Word Net-based approach for summarizing. Word-net is used to summarise the document, creating a sub-graph. The Word Net is used to assign weights to the sub-graph's nodes in respect to the synsnet. The most popular methods for text summarization incorporate one or both of the linguistic and statistical approaches.

3. ABOUT DATASET

The CNN/Daily Mail dataset is a sizable corpus of news stories and summaries gathered for summarizing purposes. It has been frequently used to train and test summarizing models and has over 300,000 article-summary pairs. This dataset includes articles and summaries on a variety of subjects, such as politics, entertainment, sports, and more. The articles and summaries are taken from the CNN and Daily Mail news websites and other some other sources [17]. The summaries in this dataset are written by professional journalists and are typically shorter than the corresponding articles, making them ideal for training summarization models. The CNN/Daily Mail dataset has been utilized in a wide range of academic projects and has significantly advanced the field of text summarization.

4. DATA PREPROCESSING

Preprocessing is an important step in working with the CNN/Daily Mail dataset. The main objective of preprocessing is to clean and transform the raw data into a suitable presentation for further analysis or modeling. Here are some common preprocessing steps that are typically applied to the CNN/Daily Mail dataset:

4.1 Data Cleaning- This step involves removing any irrelevant or redundant information and handling any missing or incomplete data. This can include removing stop words, stemming, and lemmatizing the text, as well as removing any irrelevant characters or symbols. Figure 1 highlights elements of the Dataset.

	text	y
0	LONDON, England (Reuters) -- Harry Potter star...	Harry Potter star Daniel Radcliffe gets £20M f...
1	Editor's note: In our Behind the Scenes series...	Mentally ill inmates in Miami are housed on th...
2	MINNEAPOLIS, Minnesota (CNN) -- Drivers who we...	NEW: "I thought I was going to die," driver sa...
3	WASHINGTON (CNN) -- Doctors removed five small...	Five small polyps found during procedure; "non...
4	(CNN) -- The National Football League has ind...	NEW: NFL chief, Atlanta Falcons owner critical...

Fig. 1. Elements of the Dataset

4.2 Tokenization- Tokenization comprises breaking down the text into reduced parts such as words or sentences. This is typically done using a tokenizer that can handle different types of text, such as punctuation, numbers, and special characters.

4.3 Text Normalization- Text normalization encompasses transforming the text into a standardized format [18]. This can comprise altering text to lowercase, removing diacritics,

and converting contractions to their full forms. Figure 2 highlights the Word Frequency of the Dataset.

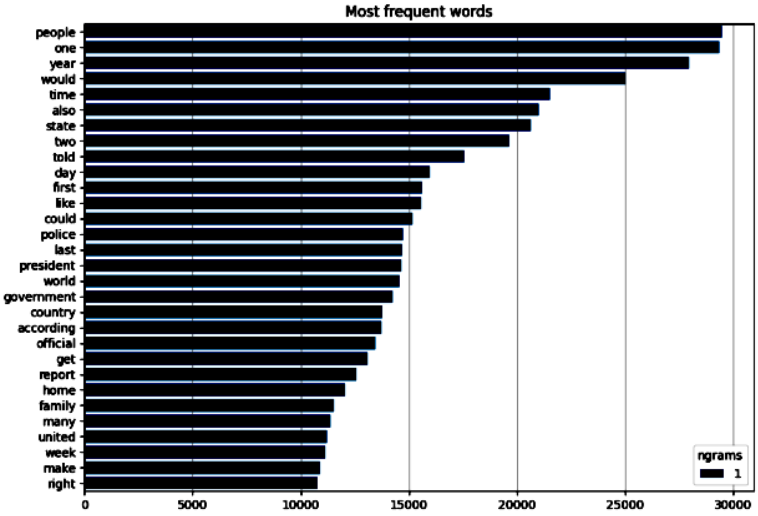

Fig. 2. Word frequency

4.4 POS Tagging- Part-of-speech (POS) tagging entails assigning the appropriate part of speech to each word in the text. This can aid in determining the text's syntactic structure and be helpful for later tasks like sentiment analysis and named entity recognition. Figure 3 highlights the Length Analysis of the Dataset.

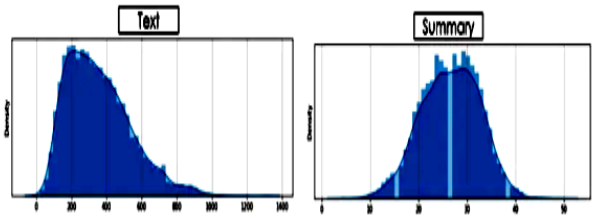

Fig. 3. Length analysis.

5. TYPES OF SUMMARIZATION TECHNIQUES

5.1 Extraction-based Summarization- The process of extraction-based summarization is locating and extracting the key expressions or sentences from the input text in order to produce a summary. The ability to retain the text's original words and meaning is one of the key benefits of extraction-based summarization, which might be crucial in certain situations [19]. Extraction-based summarization can be useful for summarizing text that is mostly factual in nature, like news items, and is also rather simple to put into practice.

Implementing extraction-based summarization can be done in several ways, including frequency-based methods and machine learning-based methods. The most significant sentences are determined using statistical criteria such as word frequency or sentence length using frequency-based approaches. Machine learning-based approaches entail building a model that can recognise the most significant sentences based on a variation of characteristics, as well as sentence length, placement in the text, and the presence of

essential words or phrases. Because the extracted sentences could not flow naturally together, extraction-based summarising has the potential to yield summaries that lack coherence and organisation. Moreover, more complicated texts that demand in-depth comprehension and interpretation may be difficult for extraction-based summarization to handle.

5.2 Abstraction-based Summarization- A technique called abstraction-based summarization includes creating a summary that does not have to match the text's exact language but rather captures the important ideas and concepts in a broader sense. To do this, natural language generation techniques are used to generate new sentences that more clearly and concisely express the major concepts of the given text. Technical or scientific texts are highly suited for abstraction-based summarization since it may capture more complicated ideas and links between concepts [20]. Abstraction-based summarization has the drawback of possibly requiring more training data and computer resources than extraction-based summarization. Also, the effectiveness of the natural language generation techniques used, which can be difficult to optimize, may have a significant impression on the quality of the summary.

6. ALGORITM SELECTION

The algorithm selection section is a crucial part of the text summarization project as it determines the performance of the model. The primary objective of this section is to gauge different machine learning algorithms and select the best performing one for the given task. In this section, we discuss the various models we considered and the criteria we used to select the best one.

•TextRank: Preprocessing the input text by removing stop words, stemming, and lower-casing the text. Imagine the text as a network, with the nodes standing in for sentences and the edges signifying how similar they are. using the graph's PageRank algorithm to isolate the summary's most crucial phrases.

•Seq2Seq: Preprocessing the input and target text by tokenizing and converting them into a numerical representation. Creating a sequence-to-sequence model with an encoder-decoder architecture and attention mechanism. Training the model on a dataset of input and target summaries. Evaluating the model performance by calculating metrics such as ROUGE and BLEU.

•BART: Preprocessing the input and target text by tokenizing and converting them into a numerical representation. Creating a BART model with an encoder-decoder architecture and fine-tuning it on a dataset of input and target summaries. Evaluating the model performance by calculating metrics such as ROUGE and BLEU.

Overall, the methodology for developing text summarization using TextRank, Seq2Seq, and BART involves a combination of data processing, model selection, training, evaluation, and optimization to achieve the desired level of accuracy and performance.

7. RESEARCH BACKGROUND

TextRank - The fundamental concept underlying TextRank is to visualize the text as a graph, where each node resembles to a phrase or a word, and the connections amongst nodes are represented by the edges. The PageRank algorithm determines the ranking of the nodes, and edges are weighted according to how similar the nodes they connect are. The input text is first pre-processed to eliminate stop words, punctuation, and other noise before being used to produce the graph. The edges connecting nodes are then determined based on how

semantically similar the sentences or words are to one another [21]. Cosine similarity, Jaccard similarity, or other metrics can be used to determine how similar two nodes are to one another.

The PageRank algorithm is used to assign a ranking to each node after the graph has been created. The PageRank algorithm is altered for TextRank to consider how similar nodes and their neighbors are. In more detail, a node's ranking is established using the average of the rankings of its neighbors, weighted by how similar the nodes are to one another. The graph's top-ranked nodes are selected to create the summary for text summarization. This can be accomplished by choosing the top-ranked expressions or words to create a summary that encapsulates the text's core concepts. Figure 4 highlights the Flow Diagram of the TextRank.

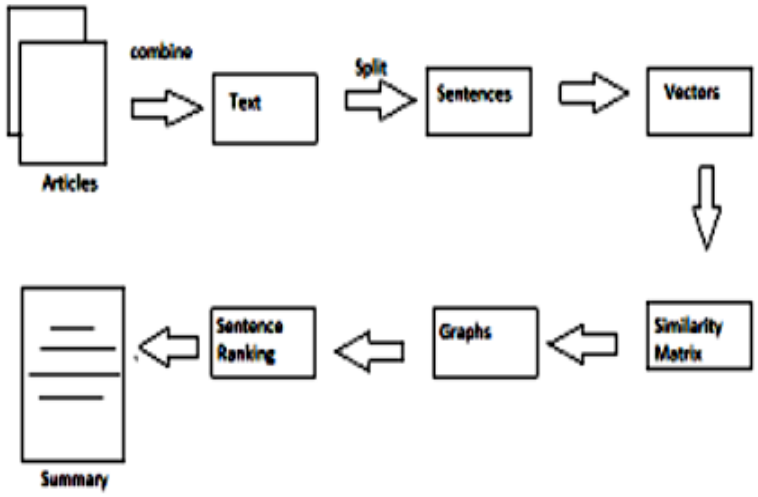

Fig.4 Flow Diagram of the TextRank

Seq2Seq - Seq2Seq models' central tenet is to discover a mapping between sequences of input and output data, such as a source language's word order and a target language's word order. Encoders and decoders are the two primary parts of Seq2Seq models. The input sequence must be processed and encoded into a fixed-length vector representation by the encoder. The decoder then receives this vector and uses the encoded input and previous outputs to construct the output sequence, one token at a time.

A recurrent neural network (RNN), such as an LSTM network or a gated recurrent unit (GRU) network, serves as the encoder in most cases. At each time step, the RNN updates its hidden state as it goes over the input sequence, single token at a time. The input sequence is represented in encoded form by the RNN's final hidden state. Although it often has a different design than the encoder, the decoder is likewise an RNN [22]. It generates the subsequent token in the output sequence using the encoded representation of the input sequence as well as the previous token that was generated as input. Up until it encounters an end-of-sequence token or a predetermined maximum length, the decoder keeps producing output tokens.

BART - On the transformer architecture, BART is based. Transformers are a type of neural network that models the connections between various elements of a sequence, such as the words in a sentence, by using attention mechanisms. An auto-regressive decoder and a bidirectional encoder are added as part of BART's extension of the transformer design. The

Seq2Seq model's encoder and the bidirectional encoder are comparable. It converts a string of tokens, like the words in a phrase, into a fixed-length vector representation as input. The BART encoder, in contrast to a typical Seq2Seq encoder, is bidirectional, which means that it processes the input sequence both forward and backward [23]. It has been demonstrated that doing so enhances performance on NLP tasks by enabling the encoder to record more intricate relationships between the tokens.

A Seq2Seq model's decoder and BART's auto-regressive decoder are comparable. It produces the subsequent token in the output sequence with the encoded representation of the input sequence as well as the previous tokens that were generated. The BART decoder, in contrast to a typical Seq2Seq decoder, is auto-regressive, which means that it generates the output tokens one at a time dependent on the tokens that were generated earlier. As a result, the decoder can recognise dependencies between output tokens and produce text that is more fluid and cohesive. A denoising autoencoder aim is used to pre-train BART using a sizable corpus of text data. By randomly masking words or rearranging expressions in the input text, the pre-training process tampers with the text before training the model to restore the original content. The model gains a broad understanding of natural language from this pre-training target, which it can then hone for particular NLP tasks. Figure 5 highlights the Flow Diagram of the BART.

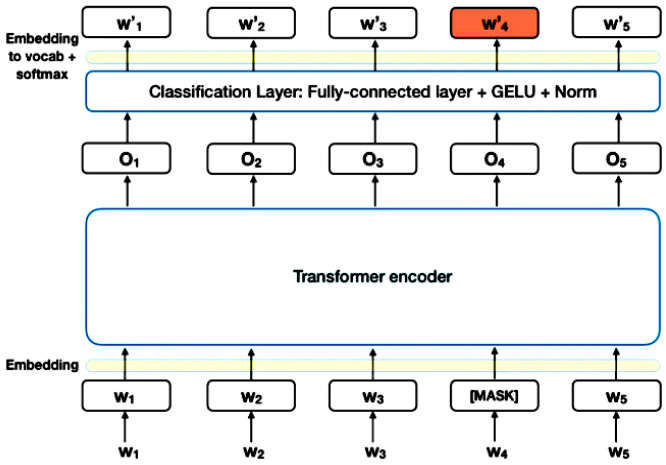

Fig.5 Flow Diagram of the BART

8. RESULTS

The results has been summarized below using TextRank, Bart, and Seq2Sq algorithms.

After almost dissipating on august 10, irene peaked as a category 2 hurricane on august 16. Irene persisted for 14 days as a tropical system, the longest duration of any storm of the 2005 season. Irene lasted for 14 days as a tropical system, the longest duration of any storm of the 2005 season. Hurricane irene began as a cape verde storm. Some of the models predicted that irene would make landfall in north carolina, while others continued to anticipate that irene would dissipate. hurricane irene was a long-lived cape verde hurricane during the 2005 atlantic hurricane season. A vigorous tropical wave moved off the west coast of africa on august 1, initially weakening due to cooler sea surface temperatures. However, the hurricane generated strong waves and increased the risk of rip currents along the east coast of the United States.

Fig 6. Predicted Summary comparison using TextRank

Figure 6 highlights the comparison between Predicted Summary and Real Summary Using TextRank.

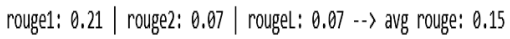

Fig 7. ROUGE Score of TextRank

Figure 7 highlights the Rouge Score for the TextRank Algorithm. ROUGE basically measure the similarity between machine-generated summaries and human-written reference summaries.

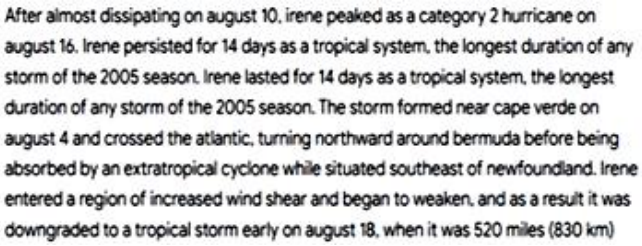

Fig 8. Predicted Summary comparison using Seq2Seq

Figure 8 highlights the comparison between Predicted Summary and Real Summary Using Seq2seq Algorithm

Fig 9. ROUGE Score of Seq2Seq

Figure 9 highlights the Rouge Score for the Seq2Seq Algorithm.

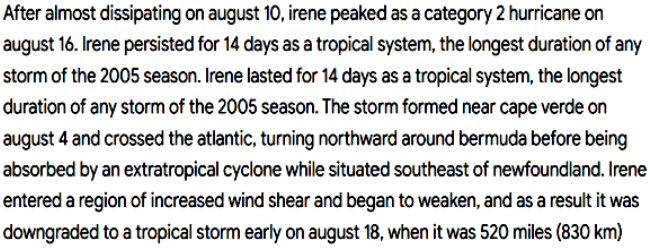

Fig 10. Passage with Predicted Summary comparison using TextRank

Figure 10 highlights the comparison between Predicted Summary and passage from dataset Using BART algorithm.

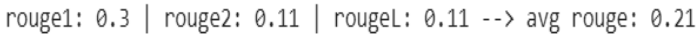

Fig 11. ROUGE Score of BART

Figure 11 highlights the Rouge Score for the BART Algorithm.

9. CONCLUSION AND FUTURE SCOPE

In conclusion, text summarizing is a crucial task in natural language processing that seeks to produce a short and cohesive summary of a lengthy text. Text summarizing can be done in a variety of ways, including with abstraction- and extraction-based methods. But recent

developments in deep learning models, including TextRank, Seq2Seq, and BART, have produced encouraging outcomes for text summarization. A graph-based system called TextRank employs the PageRank algorithm to determine the key sentences in a file and produce a summary. To create abstractive summaries, Seq2Seq models—neural network models—can be trained on pairs of documents and summaries. Contrarily, BART is a pre-trained transformer model that has attained cutting-edge performance on a variation of NLP applications, including text analysis. These methods each have advantages and disadvantages of their own. TextRank is an easy-to-use technique that works well without training data, although it might have trouble producing summaries that are fluent and cohesive. Seq2Seq models have the ability to produce abstractive summaries that are fluent and seem more natural than extractive approaches, but they need a lot of training data and can occasionally produce generic summaries. Although BART is a strong and flexible model that can produce abstract as well as cohesive summaries, it is computationally expensive and necessitates pre-training on a substantial amount of data.

Overall, the method for text summarization that is chosen depends on the particulars of the work at hand, including the volume and complexity of the input text, the length and format of the summary that is sought, and the accessibility of training data. But these recent developments in deep learning models have created new chances for more successful and effective text summarization, with the potential to be useful for a variety of applications in business and academics.

There is a large portion of scope for future development and use in the arena of text summarization using deep learning models like TextRank, Seq2Seq, and BART. Multi-document summarization, which involves summarizing several texts on the same subject, is one promising area of future research. For specialised topics, creating domain-specific summarization models is another possibility. Users' feedback on the content and presentation of summaries might be collected through interactive summarizing, resulting in more accurate and individualized summaries. Other significant research areas include the creation of efficient evaluation measures and multilingual summarization algorithms. We can anticipate more complex and successful summarization models that serve a variety of sectors and applications as long as innovation and research in this area continue.

COMPETING INTERESTS

Authors have declared that no competing interests exist.

REFERENCES

1. S. Gholamrezazadeh, M. A. Salehi, and B. Gholamzadeh, “A Comprehensive Survey on Text Summarization Systems.” 2009 2nd International Conference on Computer Science and its Applications, 2009, doi: 10.1109/csa.2009.5404226.
2. J. Goldstein, M. Kantrowitz, V. Mittal, and J. Carbonell, “Summarizing text documents.” Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, 1999, doi: 10.1145/312624.312665.
3. H. Saggion and G. Lapalme, “Generating Indicative-Informative Summaries with SumUM.” Computational Linguistics, vol. 28, no. 4, pp. 497-526, 2002, doi: 10.1162/089120102762671963.
4. H. P. Luhn, “The Automatic Creation of Literature Abstracts.” IBM Journal of Research and Development, vol. 2, no. 2, pp. 159-165, 1958, doi: 10.1147/rd.22.0159.

5. P.- ying Zhang and C.- he Li, “Automatic text summarization based on sentences clustering and extraction.” 2009 2nd IEEE International Conference on Computer Science and Information Technology, 2009, doi: 10.1109/iccsit.2009.5234971.
6. Barzilay, R., Elhadad, M, ”Using Lexical Chains for Text Summarization.” In Proc. ACL/EACL’97 Workshop on Intelligent Scalable Text summarization, Madrid, Spain,1997, pp. 10–17.
7. Y. Ko and J. Seo, “An effective sentence-extraction technique using contextual information and statistical approaches for text summarization.” Pattern Recognition Letters, vol. 29, no. 9, pp. 1366-1371, 2008, doi: 10.1016/j.patrec.2008.02.008.
8. E. Hovy and C.-Y. Lin, “Automated text summarization and the SUMMARIST system.” Proceedings of a workshop on held at Baltimore, Maryland October 13-15, 1998 -, 1996, doi: 10.3115/1119089.1119121.
9. J. Morris and G. Hirst, “The Subjectivity of Lexical Cohesion in Text.” The Information Retrieval Series, pp. 41-47, doi: 10.1007/1-4020-4102-0_5.
10. H. G. Silber and K. F. McCoy, “Efficiently Computed Lexical Chains as an Intermediate Representation for Automatic Text Summarization.” Computational Linguistics, vol. 28, no. 4, pp. 487-496, 2002, doi: 10.1162/089120102762671954.
11. Kedar Bellare, Anish Das Sharma, Atish Das Sharma, Navneet Loiwal and Pushpak Bhattacharyya, “Generic Text Summarization Using Word net. Language Resources Engineering Conference.”
12. J. Leskovec, M. Grobelnik, N. Milic-Frayling, “Extracting Summary Sentences Based on the Document Semantic Graph.” Microsoft Research, 2005.
13. Karel Jezek and Josef Steinberger, “Automatic Text Summarization (The state of the art 2007 and new challenges),” Znalosti, pp. 1-12, 2008.
14. M. Halliday and R. Hasan, “Cohesion in English.” 2014, doi: 10.4324/9781315836010.
15. Ruqaiya Hasan, Coherence and Cohesive Harmony, “In: Flood James (Ed.), Understanding Reading Comprehension: Cognition, Language and the Structure of Prose.” Newark, Delaware: International Reading Association, pp. 181-219, 1984.
16. W. C. Mann and S. A. Thompson, “Relational propositions in discourse.” Discourse Processes, vol. 9, no. 1, pp. 57-90, 1986, doi: 10.1080/01638538609544632.
17. W. C. MANN and S. A. THOMPSON, “Rhetorical Structure Theory: Toward a functional theory of text organization.” Text - Interdisciplinary Journal for the Study of Discourse, vol. 8, no. 3, 1988, doi: 10.1515/text.1.1988.8.3.243.
18. J. Morris and G. Hirst, “The Subjectivity of Lexical Cohesion in Text.” The Information Retrieval Series, pp. 41-47, doi: 10.1007/1-4020-4102-0_5.
19. R. Barzilay, K. R. McKeown, and M. Elhadad, “Information fusion in the context of multi-document summarization.” Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics -, 1999, doi: 10.3115/1034678.1034760.
20. Branimir Boguraev and Christopher Kennedy, “Salience-based Content Characterization of Text Documents,” In Proceedings of the ACL'97/EACL'97 Workshop on Intelligent Scalable Text Summarization, 1997.
21. Li Chengcheng, “Automatic Text Summarization Based On Rhetorical Structure Theory,” International Conference on Computer Application and System Modeling (ICCASM), vol. 13, pp. 595-598, October 2010.
22. Hongyan Jing, “Sentence Reduction for Automatic Text Summarization,” In Proceedings of the 6th Applied Natural Language Processing Conference, Seattle, USA, pp. 310-315, 2000.

23. Kevin Knight and Daniel Marcu, “Statistics-Based Summarization Step One: Sentence Compression,” In Proceeding of the 17th National Conference of the American Association for Artificial Intelligence, pp. 703-710, 2000.
24. Hongyan Jing and Kathleen R. McKeown, “Cut and Paste Based Text Summarization,” In Proceedings of the 1st Meeting of the North American Chapter of the Association for Computational Linguistics, Seattle, USA, pp. 178-185, 2000.
25. K. D. Garg, V. Khullar and A. K. Agarwal, "Unsupervised Machine Learning Approach for Extractive Punjabi Text Summarization," 2021 8th International Conference on Signal Processing and Integrated Networks (SPIN), Noida, India, 2021, pp. 750-754, doi: 10.1109/SPIN52536.2021.9566038.
26. K. Lanyo and A. Wausi, "A Comparative Study of Supervised and Unsupervised Classifiers Utilizing Extractive Text Summarization Techniques to Support Automated Customer Query Question-Answering," 2018 5th International Conference on Soft Computing & Machine Intelligence (ISCMI), Nairobi, Kenya, 2018, pp. 88-92, doi: 10.1109/ISCMI.2018.8703237.
27. P. Sharma and M. Chen, "Text Summarization and Keyword Extraction," 2023 14th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI), Koriyama, Japan, 2023, pp. 369-372, doi: 10.1109/IIAI-AAI59060.2023.00078.
28. S. R. R, M. R. L, D. S, A. D. P K and A. M. S, "Detection and Summarization of Honest Reviews Using Text Mining," 2022 8th International Conference on Smart Structures and Systems (ICSSS), Chennai, India, 2022, pp. 01-05, doi: 10.1109/ICSSS54381.2022.9782167.