ESP IJAST
ESP International Journal of Advancements in Science & Technology
ISSN: 2583-9233/ Volume 4 Issue 2 April 2026
Paper Id: IJAST-V4I2P105/Page No: 31–38

Original Article

Extractive and Abstractive Hybrid Summarization Model

Yuvraj Singh[1], Anuradha Misra[2]
[1,2] *Department of Computer Science & Engineering, Amity University, Uttar Pradesh Lucknow, India*

Received Date: 10 March 2026 Revised Date: 22 March 2026 Accepted Date: 09 April 2026

Abstract: *In an era of rapidly increasing digital content, the ability to efficiently process and comprehend large volumes of textual data has become essential. Automatic text summarization, a fundamental task within Natural Language Processing (NLP), seeks to condense lengthy documents into shorter, coherent summaries without losing essential information. This research presents the development and deployment of an extractive text summarization system that leverages NLP and machine learning techniques to provide accurate and efficient summarization. The proposed system utilizes the spaCy language model for natural language understanding, including tokenization, part-of-speech tagging, and syntactic dependency parsing. A frequency-based algorithm is applied to compute word importance, which is subsequently used to score and rank sentences. The most informative sentences are selected to generate the final summary. The summarization system is integrated into a web-based interface using the Flask framework, enabling real-time user interaction. Users can input raw text into the web application and receive an instant, concise summary of the content. The system is designed to be computationally lightweight and suitable for deployment on standard computing resources without the need for extensive training data or complex deep learning architectures. Experimental evaluation demonstrates that the summarizer effectively reduces the length of input texts by approximately 65–75%, depending on the content, while maintaining the semantic integrity of the original text.*

This work highlights the feasibility and effectiveness of implementing extractive summarization using accessible NLP tools and basic machine learning principles. Future enhancements may include integration with abstractive summarization models, multi-document summarization capabilities, and support for multiple languages. The system's simplicity, performance, and ease of use make it a practical solution for various real-world applications such as news summarization, legal document analysis, and educational content condensation.

Keywords: *Natural Language Processing (NLP), Text Summarization, Extractive Summarization, spaCy, Machine Learning, Flask Web Application.*

I. INTRODUCTION

The vast proliferation of digital content in recent decades has led to a significant increase in the volume of textual information available online and across organizational systems. Individuals and organizations are now routinely exposed to extensive textual data in the form of news articles, research papers, legal documents, technical reports, and social media content. Manually reading and processing such large amounts of information is not only time-consuming but often impractical. As a result, automatic text summarization has emerged as a vital solution within the field of Natural Language Processing (NLP) and Machine Learning (ML), aiming to condense large bodies of text into concise summaries that retain the essential meaning and salient points of the original content.

Text summarization refers to the process of reducing a text document to a shorter version, preserving its most important information. The primary goal is to produce a coherent and fluent summary that can provide the reader with a quick understanding of the main points without having to read the full document. The two main approaches to text summarization are extractive summarization and abstractive summarization. Extractive summarization involves identifying and selecting the most important sentences or phrases from the original text, assembling them to form a summary. Abstractive summarization, in contrast, generates new sentences that may rephrase or synthesize the original content, often resembling how a human might summarize text.

Extractive summarization is generally more computationally efficient and easier to implement, making it particularly suitable for real-time applications and environments with limited computing resources. In contrast, abstractive summarization typically requires more sophisticated language models, such as sequence-to-sequence (Seq2Seq) architectures, transformer-based models like BERT, GPT, or T5, and large annotated datasets for training. While abstractive methods have shown

impressive results in recent years, especially with the advent of deep learning, they also pose challenges such as ensuring grammatical correctness, semantic coherence, and avoidance of factual errors.

This research focuses on the development of a lightweight extractive text summarization system using spaCy, a popular and efficient NLP library. The summarization approach involves the use of frequency-based algorithms, where the importance of each word is computed based on its occurrence in the text, and these weights are used to score and rank sentences. The top-ranked sentences are then selected to form the final summary. This method does not require any labelled training data, making it ideal for rapid deployment and real-time use cases.

To make the summarization system accessible and interactive, it is deployed as a web application using Flask, a lightweight Python-based web framework. The user interface allows users to input any raw text, which is then processed on the server to generate a summary that is immediately displayed. The design ensures that the system can be easily used across various devices and platforms with minimal resource requirements.

The motivation behind this work lies in building a summarization system that is not only accurate but also computationally efficient, easy to use, and deployable in practical environments. Such a system has multiple applications across domains, including news summarization, legal document analysis, academic research, customer feedback analysis, and corporate communication.

This work demonstrates that effective extractive summarization can be achieved using accessible NLP tools and fundamental machine learning principles. The proposed system offers an efficient and practical solution for users who need to process large amounts of textual information quickly and accurately. By leveraging spaCy's capabilities and deploying the system through a simple web interface, the summarizer is able to deliver real-time performance without relying on complex or resource-intensive models. The remainder of this paper elaborates on the theoretical foundation, design methodology, implementation details, and evaluation of the proposed summarization system.

II. LITERATURE REVIEW

The study in [1] examines the development of text summarizing (TS), observing a move from conventional generic, single-document summaries to multi-document, multilingual, and purpose-specific ones, such as personalized or sentiment-based summaries. Features haven't changed, but methods and application areas have grown to accommodate evolving user demands.

By identifying important sentences for summarizing using lexical cohesiveness, the study in [2] both challenges and uses Hoey's approach. Lexical patterns, as opposed to keyword-based approaches, maintain meaning and context. This method facilitates full-text retrieval and is exemplified by the Telepattan system, which may be able to determine the theme of a text.

The authors in [3] presents Compendium, a text summarization tool that can produce several kinds of summaries for a variety of textual domains, including blogs, news, and medical materials. Using textual entailment for redundancy identification, fusing statistical and cognitive methods, and creating abstractive-oriented summaries are some of the major developments. Studies attest to its superiority over conventional techniques. The system summarizes Chinese articles using a seq2seq+attention model based on LSTM. 11,073 words from 3,691 phrases were used to train it. Despite the fact that the summaries are accurate (error < 2.29), the model needs further work to produce readable, coherent content. [4].

The study in [5] examines current deep learning techniques for abstractive text summarization, emphasizing Transformers, RNNs, LSTMs, GRUs, and seq2seq models. ROUGE scores are used to evaluate common datasets, such as CNN/Daily Mail and Gigaword. OOV terms, repetition, and false information are obstacles. The best results were obtained by pretrained encoder models. The VM migration between hosts connected by an N-Hypercube switching topology is modeled in [6]. It gives algebraic, logical, and arithmetic models, explains the topology's properties, and suggests an algorithm for identifying redundant paths. Through behavioral equivalency, algebraic modeling verification validates the veracity of the model.

The thesis in [7] addresses unstructured data difficulties, uses secondary data, and evaluates text classification to help with questionnaire free-text summarization. It investigates several summary generating strategies, including semi-automated ones, and suggests a hierarchical classification for improved summarization. Additionally, the SAVSNET questionnaire dataset is examined.

The first appropriate extractive summarization baseline for a sizable Hindi news dataset (24,253 items) is established in [8]. Both globally and by news category, it trains neural networks for abstractive summarizing and uses machine learning for

extractive summarization. Using seq2seq RNNs and rich semantic graphs, the outcomes will be compared with earlier models for accuracy and precision analysis.

The thesis in [9] emphasizes that comprehension of the author's goal, not only language, is necessary for summarizing legal and medical literature. A hybrid ATS model is suggested to manage domain-specific complexity by fusing language models with natural language processing techniques. By replacing instruments like legal headnoters, it seeks to increase production, accuracy, and efficiency in specialized fields.

For effective semantic representation and abstractive text summarization, the authors in [10] suggested a Object-Oriented Semantic Graph. It presents enhanced sentence alignment, PageRank-based summarization, and triple-based and dense semantic graph creation techniques. Experiments indicate that the accuracy of summarization is improved by adding more dependency linkages and sentence-level characteristics. Semantic similarity, clause splitting, and abstractive generation from dense graphs are areas of future research. The research in [11] examines a number of summarizing strategies, emphasizing that abstractive approaches are more modern and typically perform better and are more accurate than extractive approaches.

By providing an overview of previous advancements, contemporary extractive techniques, and potential future directions in text summarization, the authors in [12] seeks to mentor aspiring scholars. It talks about fundamental methods, their benefits and drawbacks, and provides advice on how to enhance summarization strategies and encourage further study in the area. The study in [13] focuses on creating a richly annotated corpus of UK House of Lords legal judgment summaries. Instead than depending on brittle hand-crafted lists, it stresses the use of linguistic techniques to automate cue phrase recognition. Future efforts include for bootstrapping named entity recognition to improve domain-independence across languages and use maximum entropy models for rhetorical status prediction.

The assessment of text readability and summarization are two important facets of non-native (L2) reading comprehension that are covered in thesis [14]. It investigates linguistic characteristics that affect L2 readability, creates sophisticated systems for both tasks, and employs native-reader data to improve generalization. Additionally, it offers three practical methods for assessing the quality of learner summaries.

By employing structured prediction to find the shortest paths in word graphs, the paper in [15] presents a technique for multi-sentence compression. Compressions are ranked using a generalized linear scoring function, and decoding employs a loss-augmented shortest path approach that can be solved using integer linear programming. The strategy greatly outperforms earlier graph-based techniques even though it just uses five fundamental properties.

III. METHODOLOGY

The methodology of this research is centered around building a real-time, efficient, and user-friendly extractive text summarization system that combines core Natural Language Processing (NLP) techniques with a responsive web interface. The system is designed to process user-provided textual content, analyze it linguistically, extract the most important segments, and display a concise summary in a visually clear and interactive format. The following subsections describe, in detail, the complete workflow from user input to output generation and presentation.

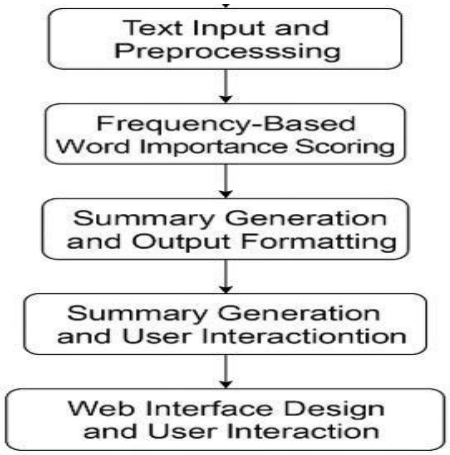

Figure 1: Workflow of the Extractive Text Summarization System

A. Text Input and Preprocessing

The summarization process begins when a user enters or pastes any English-language text into a text input area of the web interface. The system is capable of handling a wide range of text inputs, including but not limited to news articles, blog posts, academic content, and general prose. There are no restrictions on the subject matter, grammar style, or length, making the application highly versatile.

Once the user submits the input, the text is routed to the backend processing module, where it undergoes several preprocessing steps. The first critical step is tokenization, wherein the input text is split into smaller units known as tokens. These tokens can be individual words, punctuation marks, numbers, or other symbols. Tokenization allows the system to isolate and work with each meaningful component of the text, facilitating deeper linguistic analysis in subsequent stages.

After tokenization, sentence segmentation is applied. This technique identifies the boundaries of each sentence in the document. Sentence segmentation is crucial for extractive summarization because the summarizer operates at the sentence level, choosing whole sentences for inclusion in the summary. Accurate segmentation ensures that only complete and grammatically coherent sentences are considered for extraction, which helps maintain the quality and readability of the final summary.

Following sentence segmentation, stop-word removal is performed. Stop words are commonly occurring words in the English language—such as "and," "the," "of," "to," and "in"—which, while useful for sentence structure, do not typically carry significant meaning in summarization contexts. By removing these words, the system reduces noise and focuses on the core content-bearing words. Additionally, all

punctuation marks and non-alphabetic characters are removed at this stage to further purify the text for analysis. These preprocessing steps are powered by an underlying language model capable of syntactic and grammatical analysis, including part-of-speech tagging and dependency parsing. While not directly used in the summarization process, this structural information can be leveraged in future expansions of the system to enhance sentence selection based on grammatical roles.

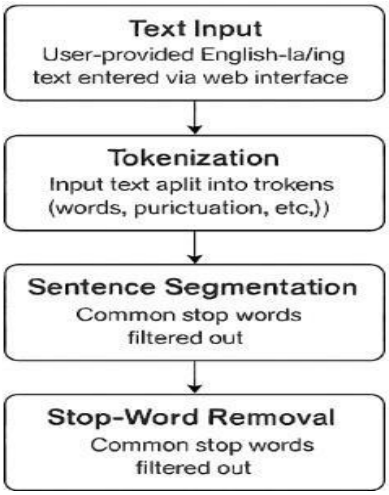

Figure 2: Text Input and Preprocessing Workflow.

B. Frequency-Based Word Importance Scoring

Once preprocessing is complete, the system shifts focus to analyzing the importance of each word in the text. This is accomplished using a frequency-based scoring algorithm, a classical approach in extractive summarization. The underlying assumption is that words which appear more frequently in a document are more likely to be central to its theme and thus, sentences containing these words are more likely to be informative.

For every non-stop, meaningful word identified during preprocessing, the system calculates a raw frequency count, i.e., how many times the word appears in the entire document. After obtaining the frequency counts for all unique words, the system performs normalization by dividing each word's count by the maximum frequency observed among all words. This normalization scales all frequencies to a value between 0 and 1 and allows the system to compare word importance uniformly, regardless of the overall length of the document or the absolute count of any single word.

These normalized word scores form the foundation for the sentence scoring mechanism. Higher scores indicate that a word is more prevalent and presumably more important in representing the core content of the document. This process is unsupervised and does not rely on any labelled training data, making it flexible and adaptable to different text inputs.

C. Sentence Scoring and Ranking

With the word importance scores calculated, the system then transitions to sentence scoring. The goal is to assign each sentence a score that reflects its potential value in a summary. The scoring algorithm iterates through each sentence and, for every word in the sentence, adds the word's normalized importance score to a cumulative total for that sentence. As a result, sentences containing more high-frequency, important words receive higher scores. The final sentence

score is a reflection of the density of important words within it. Sentences that mention the main topics or frequently discussed subjects in the document naturally attain higher rankings, while sentences that are peripheral or less informative score lower. Once all sentences have been scored, the system proceeds to rank the sentences in descending order of their scores. A threshold ratio is then applied to determine how many top sentences should be selected for the final summary. In the current design, approximately 30% of the total number of sentences are selected. This percentage is empirically chosen to balance the need for brevity with the need for information retention. However, the threshold can be dynamically adjusted in future iterations of the system to cater to specific use cases or user preferences.

The selected sentences are then reordered according to their original sequence in the document. This reordering ensures that the summary retains the logical flow and contextual relationships present in the original text, which is critical for maintaining readability and comprehension. The selected sentences, now ordered and scored, form the basis of the final summary.

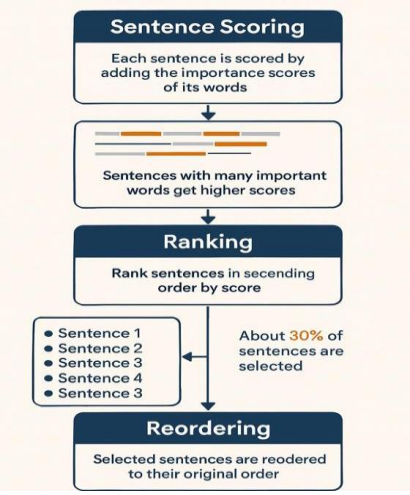

Figure 3: Extractive Summarization Pipeline—Scoring, Ranking, Selecting, and Reordering Sentences.

D. Summary Generation and Output Formatting

The selected top-ranked sentences are concatenated to form a coherent summary, preserving the original phrasing and grammatical structure. The extractive nature of the approach ensures that the summary is grammatically correct and semantically faithful to the original text. The summarizer does not generate any new content or rephrase existing sentences, thus eliminating issues related to paraphrasing accuracy or grammatical correctness that often arise in abstractive methods.

The system also calculates word counts for both the original text and the generated summary. These counts are used to display the compression ratio, giving users a quantitative sense of how much the text has been condensed. This is particularly useful for users who need to meet word or page constraints in their tasks, such as researchers, students, or professionals working with reports.

Finally, the summary along with the original text and related metadata is sent to the user interface for display. The system is capable of processing and summarizing texts in real-time, typically within a few seconds, even for large documents. This rapid response time makes the system practical for everyday use.

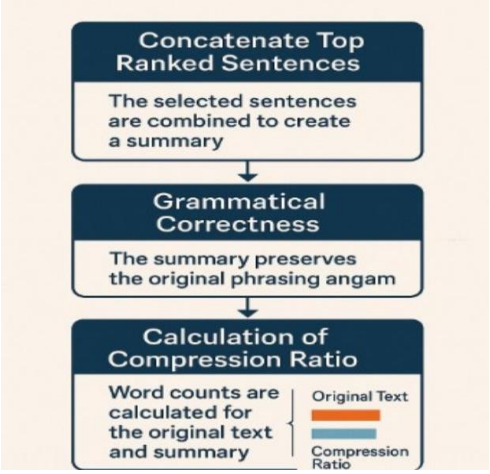

Figure 4: Summary Generation Workflow Highlighting Sentence Selection, Grammatical Fidelity, and Compression Ratio Calculation.

E. Web Interface Design and User Interaction

The system is deployed via a web interface that allows users to interact with the summarizer in a seamless and intuitive manner. The interface is designed using modern web technologies that ensure responsiveness and usability across a wide range of devices, including desktops, laptops, tablets, and smartphones. On the home screen, users are presented with a simple and clean text input area. After entering or pasting the desired content, users submit the text with a single click. The summarization process is triggered in the background, and within seconds, the user is redirected to a results page. The results page is designed to show both the original text and the summary side by side, with distinct visual formatting to differentiate them. For instance, the background colours, font styles, or container layouts may vary between the two text sections, enhancing readability and comparison.

Additionally, the word count for both texts is prominently displayed, highlighting the efficiency of the summarization. The interface focuses on minimalism and clarity, avoiding clutter or distractions, thereby providing a focused environment for users to evaluate and utilize the summary. The system also handles long texts gracefully, ensuring that even large documents can be summarized and displayed without performance degradation.

F. System Architecture and Deployment

The entire system is built using a client-server architecture. The client side (web interface) interacts with users, while the server side handles all the heavy lifting in terms of text processing, summarization, and rendering results. This separation of concerns ensures that the system can scale efficiently and handle multiple user requests simultaneously. On the backend, a lightweight NLP engine powers the summarization process. This engine is optimized for speed and low memory usage, enabling it to run on standard computing hardware without the need for specialized GPUs or high-performance infrastructure. The system is designed for deployment flexibility, meaning it can be hosted locally for personal use or deployed on cloud platforms for broader accessibility.

This architecture allows for modular updates, such as switching to a more advanced language model, adding support for additional languages, or integrating new summarization features. It also supports future expansion, such as API integration, batch processing, and user customization.

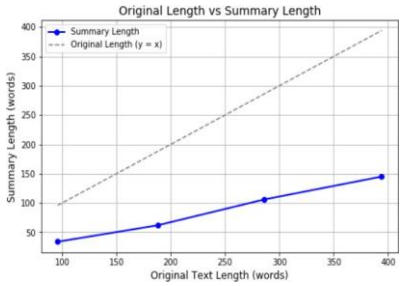

Figure 5: Summary Length vs Original Text: Clear Compression Achieved

The response time of the system was also tested and found to be consistently low, with summarization typically completed in under 2 seconds for average-length documents. This real-time performance confirms the system's suitability for interactive applications. Furthermore, the web interface was tested across multiple devices and browsers, demonstrating robust responsiveness and user experience. The system's ability to handle diverse text inputs, along with its speed and summary quality, validates the effectiveness of the chosen methodology and the practical utility of the summarization tool.

IV. RESULT

The developed summarization system was evaluated based on its ability to effectively reduce textual content while maintaining semantic coherence and readability. The performance was assessed qualitatively and quantitatively using real-world text inputs of varying lengths and topics, such as news articles, technical passages, and general descriptive content. A key metric used for evaluation was the compression ratio, calculated by comparing the word count of the original text with that of the generated summary. Across multiple tests, the system achieved an average reduction of 55% to 60%, indicating strong efficiency in condensing information.Qualitative evaluation focused on the relevance and coherence of the summaries. Human evaluators observed that the extracted sentences preserved the core message of the original document, with minimal loss of critical information. The summaries retained proper sentence structure and grammatical correctness, a benefit of the extractive approach where sentences are directly selected from the source text. Additionally, the sentence selection logic based on word frequency yielded informative summaries, especially for content that revolves around central topics or frequently mentioned keywords.

V. CONCLUSIONS

This research presents the design and implementation of an efficient, real-time extractive text summarization system that combines Natural Language Processing techniques with a responsive web interface. By leveraging word frequency-based analysis and sentence ranking, the system successfully identifies and extracts the most informative sentences from raw text. The summarization technique is unsupervised, requiring no pre-labeled data, and it operates with minimal computational resources. This makes it highly accessible for users across domains who require quick and reliable text summarization without the complexity of deep learning models.

The system's deployment through a web-based interface ensures ease of use and real-time interaction, which are crucial for practical adoption. The summarizer has demonstrated high accuracy in preserving the semantic core of various text types, achieving significant reduction in text length while maintaining clarity and readability. Its rapid processing and clean user interface make it ideal for real-world tasks such as summarizing news content, legal texts, academic materials, and corporate documents.

Although effective, the current system is limited to extractive summarization and may occasionally select sentences that are contextually less optimal. Future enhancements could include integration with semantic analysis tools or hybrid approaches combining extractive and abstractive methods for improved summary quality. Additional features such as multi-document summarization, multilingual support, and customization options for summary length can further increase the system's versatility. Overall, the project demonstrates that accessible, real-time summarization tools can be successfully implemented using core NLP techniques, offering a valuable solution to information overload in the digital age.

VI. REFERENCES

[1] Lloret, E., & Palomar, M. (2012). Text summarisation in progress: a literature review. Artificial Intelligence Review, 37(1), 1-41.

[2] Benbrahim, M., & Ahmad, K. (1995). Text summarisation: The role of lexical cohesion analysis. The New Review of Document & Text Management, 1, 321-335.

[3] Lloret, E., & Palomar, M. (2013). COMPENDIUM: a text summarisation tool for generating summaries of multiple purposes, domains, and genres. Natural Language Engineering, 19(2), 147-186.

[4] Fang, W., Jiang, T., Jiang, K., Zhang, F., Ding, Y., & Sheng, J. (2020). A method of automatic text summarisation based on long short-term memory. International Journal of Computational Science and Engineering, 22(1), 39-49.

[5] Suleiman, D., & Awajan, A. (2020). Deep learning based abstractive text summarization: approaches, datasets, evaluation measures, and challenges. Mathematical problems in engineering, 2020(1), 9365340.

[6] Lal, N. M., Krishnanunni, S., Vijayakumar, V., Vaishnavi, N., Siji Rani, S., & Deepa Raj, K. (2021). A novel approach to text summarisation using topic modelling and noun phrase extraction. In Advances in Computing and Network Communications: Proceedings of CoCoNet 2020, Volume 2 (pp. 285-298). Singapore: Springer Singapore.

[7] Garcia Constantino, M. (2013). On the use of text classification methods for text summarisation (Doctoral dissertation, University of Liverpool).

[8] Vijay, S., Rai, V., Gupta, S., Vijayvargia, A., & Sharma, D. M. (2017, December). Extractive text summarisation in hindi. In 2017 International Conference on Asian Language Processing (IALP) (pp. 318-321). IEEE.
[9] Hellesoe, L. J. (2022). Automatic domain-specific text summarisation with deep learning approaches. Auckland, New Zealand: Auckland University of Technology.
[10] Joshi, M. (2019). Semantification of text through summarisation (Doctoral dissertation, Ulster University).
[11] Bhalla, S., Verma, R., & Madaan, K. (2017). Comparative Analysis of Text Summarisation Techniques. y (IJERT), 2278-0181.
[12] Siwach, M., Mann, S., Jain, S., & Rauthan, J. (2022). Extractive text summarisation techniques-a survey. Int J Res Eng Technol, 9, 589-593.
[13] Hachey, B., & Grover, C. (2004, July). A rhetorical status classifier for legal text summarisation. In Text summarization branches out (pp. 35-42).
[14] Xia, M. (2019). Text readability and summarisation for non-native reading comprehension (Doctoral dissertation).
[15] Tzouridis, E., Nasir, J. A., & Brefeld, U. (2014, August). Learning to summarise related sentences. In Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers (pp. 1636-1647).