**International Journal of Applied Mathematics**

**Volume 38 No. 1s, 2025**

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

# HINDI TEXT SUMMARIZER USING ABSTRACTIVE AND EXTRACTIVE TECHNIQUE

**[1]Gurveen Kaur Bans, [2]Neelam Phadnis**

[1]ME Student, Computer Engineering Department, Mumbai University/ Shree L.R. Tiwari College of Engineering, Thane, India

[2]Assistant Professor, Computer Engineering Department, Mumbai University/ Shree L.R. Tiwari College of Engineering, Thane, India

Email: gurveen555@gmail.com, Neelam.phadnis@slrtce.in

**Abstract**

Text summarization is a significant process in retrieving the information, supporting the generation of concise summaries from original text documents. This paper looks at the importance of text summarization in many different uses those being Article Proofreading, Online Visibility Enhancement, Corporate Assessment, and Competitive Analysis. Both the two main text summarization strategies, a creative and a concise one, will be debated. Summary based on an extractive method depends on identifying important sentences and words from the text, based on high scores. On the other hand, an abstractive summary can contain some new expressions which are not presented in the unduplicated text. The paper is based on an extractive summarization model that has been developed using TFIDF. TFIDF approaches, Clustering-based techniques, and ML approaches are reported to be efficient for extractive text summarization and abstract based model is developed using BERT technique its ability to perceive language structures, sentence relationships, and to understand meaning gets significantly improved in a similar patterns of humans. Having been pre-trained, BERT will be able to do fine-tuning for numerous downstream NLP tasks, e.g., summarizing the text, NER, and Q and A. One of its strengths is being able to produce contextualized vector representations of words, where the semantical association of a word is dynamically modulated by the context. The study reviews the current approaches and models for Hindi text summarization, concentrates on the semantic relatedness principle and the keyword feature extraction.

**Keywords**: Term Frequency-Inverse Document Frequency (TFIDF), Stemming, Tokenization, Text Summarization, Natural Language Processing.

## 1. Introduction

The latter usually contains more information than is needed especially when the goal is to fully understand the information in question. This makes it difficult to gather all the information needed from a single document or a single web page. As a result, people spend a lot of time identifying and sorting through ample data for deriving meaningful information from it. To this end, document summarization has risen as a method of generating automated

Received: July 18, 2025

International Journal of Applied Mathematics
Volume 38 No. 1s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

abstracts of textual data, including the key points. This way, the method could assist clients in making decisions and locating the information they need faster by providing them with a shorter version of the text that contains all the necessary information.
Summary is a key task in several tasks such as; news editing, search engine optimization, business intelligence and market analysis among others. Therefore, it assists in avoiding information overload where people are only able to get the information they require with minimum effort as opposed to having to read through numerous papers. To this end, this paper proposes extractive summarization as a solution to the problem of information overload using the TF-IDF model. Extractive Summarizing is the process of important keyword selection and phrases from the input data and using them to create the summary, a length of the input proportional to the length of the output. It identifies and extracts the most relevant phrases and keywords from the input text. Therefore, the result is the main idea and the full value of the source material in the summary. On the other hand, abstractive summarization expresses a new representation of the content and a reinterpretation of the text, which can be seen as being similar to what a human author would have written. This can result in the introduction of words that were not present in the input at all [1].Finally, the paper highlights the growing significance of text summarization techniques in improving accessibility, decreasing cognitive load, and increasing understanding in a increasingly data rich environment.

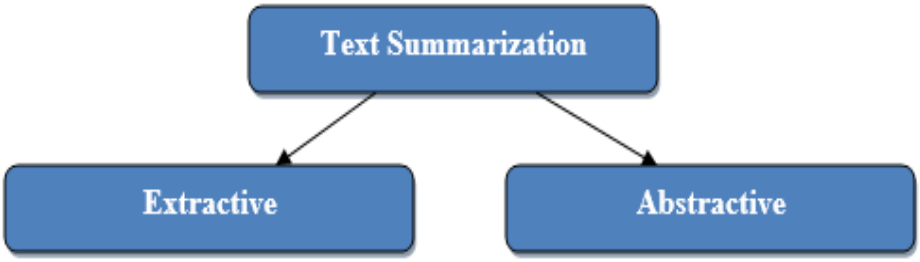


Figure 1, Summarization Techniques

**EXTRACTIVE TEXT SUMMARIZATION**

As an extractive summarizer, the main task is to find main sentences in the text and at the same time, make the outline shorter than the original text, without repeating sentences. This is done by a direct utilization of portions of the input text [2].

**Relevance - Term Frequency Inverse Document Frequency (TF-IDF) Algorithm:**

An ethically friendly method called Term Frequency Inverse Document Frequency (TFIDF) algorithm is used in information retrieval and text mining for extractive summarization.

Term Frequency (TF):
Definition: Term Frequency is simply the statistics of how significant a word is within one document.
Calculation: TF (tt, dd) = Total no. of terms(tt) in the documen(dd)/ No. of times term(tt) occurs in the document(dd) (1)

Received: July 18, 2025

International Journal of Applied Mathematics

Volume 38 No. 1s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Inverse Document Frequency (IDF):
Definition: IDF is a technique that calculates the uniqueness of a term across a document collection.
Calculation: IDF (tt, DD) = log (No. of documents containing term tt / Total no. of documents in the collection DD)

TF-IDF (Term Frequency-Inverse Document Frequency):
Calculation: TF-IDF(tt,dd,DD)=TF(tt,dd)×IDF(tt,DD) (2)
Weighting Process:
Purpose: The weighting process serves to produce weights for terms based on their scored values and makes it a more robust calculation by regarding a term in a document or in a collection of documents as important. Implementation: The TF-IDF values of the higher terms are considered more significant. The weighting process may involve using these values directly or some other transformations as required. The TFIDF is used by the model trained in an extractive summarization fashion to identify phrase relevance within the document and then selecting the most important sentences from the document based on their TFIDF scores. [9].

**ABSTRACTIVE TEXT SUMMARIZATION**

**BERT: Bidirectional Encoder Representations from Transformers.**

This model introduced by google browser in 2018 is a pre-trained language model that, a Transformer model. It has revolutionized natural language understanding by capturing linguistic context and nuances in the most efficient way. It is especially helpful for abstractive tasks. Training the model beforehand on a vast dataset of data in text form using two primary unsupervised learning tasks: MLM and NSP [12]. The training techniques allow this algorithm to construct a deep contextual understanding of words and their relationships within sentences [11] [12].

The architecture is based on the Transformer framework, this framework utilizes internal focus mechanisms in order to execute the entire input series in parallel. In contrast to classical approaches, which are reliant on a unidirectional context, BERT acquires both forward and backward dependencies, thereby, it considers not only left but also right contextual information for each word in a sentence. Thanks to this bidirectional mechanism, its ability to perceive language structures, sentence relationships, and to understand meaning gets significantly improved in a similar patterns of humans. Having been pre-trained, BERT will be able to do fine-tuning for numerous downstream NLP tasks, e.g., summarizing the text, NER, and Q and A. One of its strengths is being able to produce contextualized vector representations of words, where the semantical association of a word is dynamically modulated by the context. [11]

Owing to the powerful contextual understanding and adaptability, BERT has become a widely adopted model for abstractive summarization. Its ability to effectively grasp complex

Received: July 18, 2025

International Journal of Applied Mathematics
Volume 38 No. 1s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

linguistic nuances and generate coherent summaries has positioned it as a fundamental tool in modern natural language processing applications [11]

## 2. Literature Analysis

Table 1. Literature Review

| Authors | Title | Approach | Key Findings |
|---|---|---|---|
| Rajasekaran, Abirami and Dr R. Varalakshmi [8] | Review on automatic text summarization[8] | Supervised Machine Learning Technique[8] | Reviewed various approaches in Automatic Text Summarization. Explored methodologies in shortening text while retaining primary information content. |
| P. Janjanam and C. P. Reddy [14] | Text Summarization: An Essential Study [14] | Graphical based Strategies[14] | Surveyed graphical based strategies for extracting text summarization. This paper discusses abstractive and extractive text summarization techniques. |
| R. Boorugu and G. Ramesh [13] | A Survey on NLP based Text Summarization for Summarizing Product Reviews [13] | NLP-based Text Summarization[13] | Explored text summarization approaches for condensing product reviews into concise summaries. Discussed the importance of summarizing lengthy assessments for online consumers. |
| P. R. Dedhia, H. P. Pachgade, | Study on Abstractive Text | Abstractive Text Summarization[15] | Investigated current models for |

Received: July 18, 2025

Volume 38 No. 1s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

| | | | |
|---|---|---|---|
| A. P. Malani, N. Raul, and M. Naik [15] | Summarization Techniques[15] | | abstractive text summarization and identified potential areas for further studies. |
| N. S. Shirwandkar and S. Kulkarni[16] | Extractive Text Summarization Using Deep Learning[16] | Extractive Text Summarization[16] | For the purpose of this paper, a approach to extractive text summarizing was developed using RBM and FL. They are integrated for the selection of essential phrases to generate a meaningful and lossless summary. |

## 3. Methodology

Internet hosts an extensive collection of digital content, much of which is highly informative. However, the vast amount of available data often leads to information overload, which inturn makes it tough for endusers to extract only the most relevant details efficiently. To address this, one of the important applications in information retrieval is text summarization, which compresses lengthy text into a concise version while securing its essential meaning and information. This approach enables users to quickly access the most relevant content without sifting through excessive data.

In the vast amount of information on the Internet, the user wants to find the most valuable and important sources in the shortest time possible. Search engines help in this process by ranking web pages according to relevance, and often using algorithms such as PageRank to determine the importance of pages. The PageRank algorithm was originally developed to rank web pages, but it can be adapted for text ranking as well. By changing the focus from hyperlink-based ranking to similarity-based ranking, the PageRank-style matrix can be used to determine the relevance of text instead of just the connections between webpages. Thus, the proposed alternative approach can be used to organize and retrieve information more effectively and, therefore, is useful in the solution of a number of text processing tasks.

**Algorithm for Proposed System**

1. Term frequency and weighting
2. TFIDF sentence weighting
3. Rule-Based summarizing strategy
4. BERT: Bidirectional Encoder Representations from Transformers.

Received: July 18, 2025

**International Journal of Applied Mathematics**

**Volume 38 No. 1s, 2025**

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

*Term frequency and weighting*

Term Frequency (TF):

It measures the probability of a term occurring in a document(dd) and also facilitates the comprehension of the general organization of the information in terms of the key words.

$$TF(tt, dd) = \frac{No.of\ times\ term\ tt\ appears\ in\ document\ dd}{Total\ no.of\ terms\ in\ document\ dd}$$

Inverse Document Frequency (IDF):

The DD is a way utilized for calculating the rarity or uniqueness of a term in a corpus. It is useful to unblock the effect of the baggage of the common words and to enhance the importance of the less frequent words in a corpus:

$$IDF(tt, DD) = log\frac{Total\ no.of\ documents\ in\ the\ collection\ DD}{No.of documents\ containing\ term\ t+1}$$

TF-IDF (Term Frequency-Inverse Document Frequency):

Calculation: It is the product of TF and IDF for a single phrase within the given document. In effect, it is a function of the phrase probability in given document(dd) combined with its infrequency across all documents [10]

$$TF - IDF\ (tt, dd, DD) = TF(tt, dd) \times IDF(tt, DD)$$

Weighting Process: Purpose: The weighting process is used to determine the weight of terms depending on their TF-IDF values, which makes it a more accurate measure of the importance of a term in one or several documents. Implementation: Some examples of higher TF-IDF values include: The weighting process may involve using these values directly or applying additional transformations according to the specific needs [10].

*TFIDF sentence weighting*

The TF-IDF sentence weight process assigns weights to sentences in a document based on the relevance of the terms they contain. Such an approach is frequently employed in application areas, such as text summarization, document ranking, and information retrieval. The following is a step by step explanation of the TF-IDF sentence weighting process:

Tokenization: This is the process of breaking down a document into its word or sentence level components.

Term Frequency (TF) Calculation for Sentences: Determine the tf of each word in each sentence. This includes counting the term frequency of each word within each sentence.

$$TF(tt, ss) = \frac{No.of\ times\ term\ tt\ appears\ in\ sentence\ ss}{Total\ no.of\ terms\ in\ sentence\ ss}$$

Document Frequency (DF): It is the counting of how many sentences contain each term.

$$DF(tt) = No.of\ sentences\ containing\ term\ tt$$

Inverse Document Frequency (IDF) Calculation: Determine the IDF for every word from the corpus, where the length of the corpus in sentences and the DF.

Received: July 18, 2025

**International Journal of Applied Mathematics**

**Volume 38 No. 1s, 2025**

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

$$IDF(tt, DD) = \frac{Tot.no.of\ sentences\ in\ the\ document\ collection\ DD}{DF(tt)+1}$$

Step-by-Step Process of TF-IDF Method:-

1. Document Pre-processing:
Tokenization: This document will be segmented into single words, sentences. This is important for analysing the frequency of terms within the text.
Stop-word enumeration: Words like "and", "the", "is" that do not add to the meaning are taken out from the piece of text.
Stemming/Lemmatizing: Words are trimmed to their root structure to ensure that different forms of words are treated similar (e.g., "running "becomes "run") [4][6][7].

2. Term Frequency (TF) Calculation:
Definition: It measures the occurrence rate of a phrase that appears in the document.
Calculation: For each term *tt* in a document *dd*, TF is calculated as: TF (tt,dd)=No. of occurrences of tt in dd. Total no. of terms in dd
*TF*(*tt*,*dd*)=Total no. of terms in *dd* No. of occurrences of *tt* in *dd* .This gives a normalized frequency of the term within the document[10].

3. Document Frequency (DF) Calculation:
Definition: Keeps track about a term by counting it occurs in how many documents.
Calculation: For each term *tt*, DF is calculated as the number of documents containing *tt*.

4. Inverse Document Frequency (IDF) Calculation:
Definition: Phrase distinctiveness is calculated across a set of documents.
Calculation: IDF is calculated as:
IDF (tt, DD)= log(Total number of documents in DD. Number of documents containing tt)

5. TF-IDF Calculation:
Definition: TF-IDF combines the local significance of a term with its global importance.
Calculation: The word score *tt* in a document *dd* is computed as:
$TFIDF(tt, dd, DD) = TF(tt, dd) \times IDF(tt, DD)$ Sentence Weight Calculation: Each sentence in the document is assigned a weight based on the TF-IDF scores of the terms it contains.
Calculation: For each sentence *ss*, the overall weight is calculated as:
Weight (*ss*, *DD*)=$\sum t \in s$ *TFIDF*(*tt*, *ss*, *DD*).
This sums the scores of all words in the sentence to get its total weight.

6. Ranking Sentences:
Once all sentences have been assigned weights, they are graded based on their scores. High scores indicate more important sentences.

Received: July 18, 2025

International Journal of Applied Mathematics
Volume 38 No. 1s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Selection Criteria: A threshold can be set to select the top n*n* phrases which is to be incorporated in the executive summary [8].

7. Summary Generation:
Generation of final summary is done by combining the shortlisted sentences. This may include: Ordering: Sentences can be ordered based on their original position in the given document to maintain coherence.
Post-Processing: Optional steps may include grammatical corrections, sentence compression, and ensuring coherence between selected sentences.

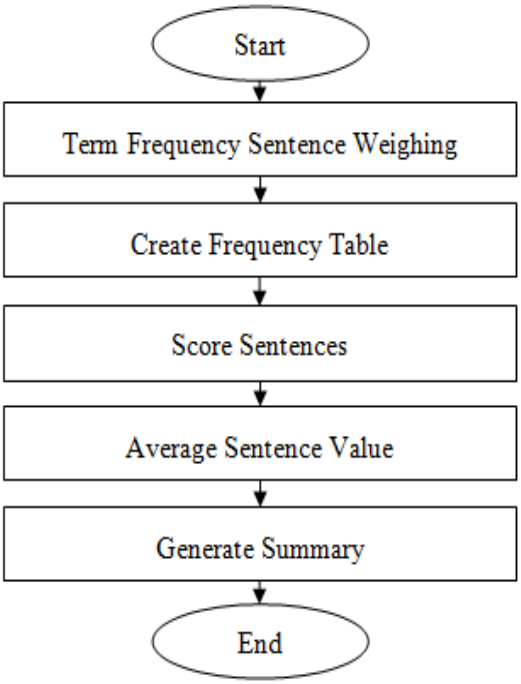


Figure 2**,** Summary Generation Process.

TF-IDF Calculation for Sentences: Multiply the TF of each term in a sentence by its IDF to get the value for that term in that sentence. TF-IDF (tt, ss, DD) = TF (tt, ss) × IDF (tt, DD)
Sentence Weight Calculation: Sum the TF-IDF values of all terms in a sentence to get the overall weight of the sentence. This can be done for each sentence in the document. Sentence Weight (ss, DD)=∑tTF-IDF(tt, ss, DD)
Ranking or Selection: Optionally, sentences can be ranked based on their weights, and a summary can be generated by selecting the top-ranked sentences.

**Rule-Based Summarization Strategy**

Tokenization: The input document is divided into tokens - words or sentences - according to the required granularity.
Sentence Extraction: Identify important sentences based on predefined rules. For example: Select sentences containing specific keywords or phrases. Prioritize sentences based on their position (e.g., the first or last sentence). Consider sentence length as a factor in importance.
Scoring Criteria: Define criteria for scoring sentences based on their importance. Criteria may include: Frequency of certain keywords or terms. Positional importance. Sentence length. Named entity recognition [8].
Summarization: Sentences scoring maximum score is selected based on defined rules and criteria. Finally, the last summary is created by putting together the chosen sentences.

Received: July 18, 2025

Volume 38 No. 1s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Optional Post-Processing: Apply post-processing steps to improve the coherence and readability of the summary. This may involve: Sentence compression to reduce redundancy. Ensuring grammatical correctness. Checking for coherence between selected sentences.

**Bidirectional Encoder Representations from Transformers (BERT)**

NLP uses bert which is basically from transformers, that has been previously trained by Google in 2018 [12]. BERT, being one of the Transformer's architectures, was a significant evolution in the NLP era and is popular these days. This robot is trained on deep learning theories to further develop NLP applications with profound contextual knowledge optimally [11]. The given example includes a brief summary of the main components and functioning principles of this approach:

1. Pre-training:

BERT undergoes pre-training on a vast corpus of textual data using two unsupervised learning objectives:

Masked Language Model (MLM): Certain words are masked by the model randomly inside a sentence, and to predict the words based on contextual clues from neighbouring words are basically learned by the model.

Next Sentence Prediction (NSP): BERT is trained to determine if a particular couple of sentences appear continuously in the original text, enhancing its understanding of sentence relationships.

2. Architecture:

Transformer Framework: Transformer architecture is used in building it that uses internal focus mechanisms in parallel to process the input sequences.

Bidirectional Context: Old NLP models relied on unidirectional processing, whereas, BERT captures dependencies from both the sides of a word in a sentence, directing towards a deeper contextual representation.

3. Tokenization:

BERT employs Word Piece Tokenization, which breaks words into sub word units. This technique allows it to properly handle huge vocabulary, including rare and compound words.

4. Model Layers:

Encoder Stack: The model comprising numerous encoders with each one of them housing self-attention mechanisms and feedforward neural networks.

Parameter Sharing: During pre-training, parameters are shared across layers, allowing the model to learn hierarchical linguistic features.

5. Fine-Tuning:

Task-Specific Fine-Tuning: After pre-training, BERT can be adapted for various NLP applications like classification of textual data, NER, Q and A.

Received: July 18, 2025

**International Journal of Applied Mathematics**

**Volume 38 No. 1s, 2025**

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Task-Specific Layers: Additional layers tailored to specific tasks can be incorporated, and the algorithm training is done on small dataset relevant to the given application.

6. Embedding Output:

This way, BERT provides such embeddings to AIs, which mean that different meanings of words or indeed, words themselves, are dependent on the context.

7. Applications:

Transfer Learning: The pre-trained BERT model can be tailored for diverse NLP operations, achieving best results even with limited task-specific training data.

8. Benefits:

Context Understanding: BERT excels in capturing context and understanding nuances in language.

Reduced Task-Specific Data Requirement: Due to pre training on a large dataset, BERT requires less task-specific training data.

9. Limitations:

Computational Intensity: BERT's architecture is computationally intensive, limiting its deployment in resource-constrained environments.

Fixed Context Window: Despite bidirectionality, BERT still has a fixed context window, and it may not capture extremely long-range dependencies effectively.

Step-by-Step Process of BERT

Step 1: Tokenization: The input text is split into tokens. BERT uses Word Piece tokenization to handle sub words and out-of vocabulary words

Input Formatting: Tokens are converted into input IDs and include special tokens ([CLS] for classification and [SEP] for separating sentences).

Step 2: Encoding the Text BERT Model Initialization:

The pre-trained BERT model is loaded, which contains layers of transformers.

Text Encoding: The pre-processed tokens are fed into BERT, which outputs contextualized embeddings for each token. This captures the relationships between words and their context.

Step 3: Feature Extraction

Sentence Embeddings: For extractive summarization, embeddings for entire sentences can be obtained by pooling (e.g., taking the average or using the [CLS] token's representation).

Semantic Representation: These embeddings represent the meaning of sentences in a high-dimensional space, allowing the model to understand which sentences are important.

Received: July 18, 2025

Volume 38 No. 1s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Step 4: Sentence Scoring and Ranking
Similarity Measurement: Using cosine similarity or other distance metrics, the model scores sentences based on their relevance to the main themes of the document.
Ranking: Ranking of the sentence is based on the scores they generate, with higher-ranked sentences are most preferred ones to be included in the summary [8].

Step 5: Summary Generation
Extractive Summarization: The best sentences are taken for the synopsis, maintaining the original wording and structure.
Abstractive Summarization: If BERT is used in an encoder and decoder setup, the encoder (BERT) processes the textual data, and the decoder generates a summary by producing originals that retain the meaning of the original content.

Step 6: Fine-tuning (if applicable)
If available, obtain finer tuning (if applicable) below.
Task-Specific Training: The model can be trained further using summarization datasets to improve its understanding of domain-specific language and enhance its summarization skills.

Step 7: Enhancement of Coherence and Fluency.
Language Modelling: BERT can also help improve the coherence and fluency of the generated summaries, particularly in the abstractive approach, by understanding the flow of natural language.

Step 8: Post-processing.
Final Refinement: The final output of the summaries can also be further processed to eliminate some redundancy or to make the output more readable.

**Model Accuracy Metrics**
Precision: A precision figure is the percentage of correctly summarized sentences to all sentences chosen by the system. A precision score of more than one indicates that the model is accurate in identifying the specific information without incorporating a number of irrelevant sentences [3].
Formula: $Precision = \frac{True+ve\ (TP)}{True+ve\ (TP)+False+ve(FP)}$

Recall: Recall is measured up to how many of the actual reference summary sentences were chosen correctly by the model. A large recall score indicates that the summarization model has captured most of the essential information [3].
Formula:$Recall = \frac{True+ve\ (TP)}{True+ve\ (TP)+False-ve\ (FN)}$

Received: July 18, 2025

International Journal of Applied Mathematics
Volume 38 No. 1s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

F1 Score: The F-score is a measure of overall performance that emphasizes the harmonic mean of precision and recall, treating the metrics equally. It is most valuable when dealing with tipped data and means that both relevance (precision) and completeness (recall) are taken into account [3].
Formula: $F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$

G-Score: It is the geometric mean of P and R, proposing a different measure for assessing the quality of summarizations [3].
Formula: $G = \sqrt{Precision \times Recall}$

Table 2. Accuracy Metrics [5]

| Metric | Validation Dataset | Test Dataset |
|---|---|---|
| F1 ROUGE-1 | 0.551466 | 0.544284 |
| F1 ROUGE-2 | 0.457683 | 0.443253 |
| F1 ROUGE-3 | 0.431636 | 0.414829 |
| F1 ROUGE-4 | 0.417699 | 0.399905 |
| PRECISION | 0.498256 | 0.489563 |
| RECALL | 0.657104 | 0.652448 |
| G – SCORE | 0.503248 | 0.487699 |

Received: July 18, 2025

**International Journal of Applied Mathematics**

**Volume 38 No. 1s, 2025**

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

## 4. Results And Discussion

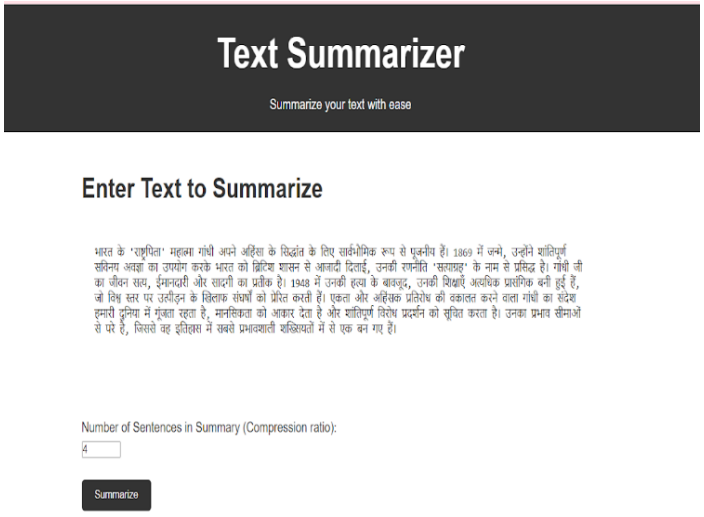


Figure 3, Summary Generation Process


Figure 4, Term Frequency and Weighting


Figure 5, TFIDF Sentence Weighting

Received: July 18, 2025

**International Journal of Applied Mathematics**

**Volume 38 No. 1s, 2025**

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)


Figure 6, Rule Based Summary


Figure 7, BERT

## 5. Conclusion

The research compares and contrasts both the summarization techniques. Extractive summarization, specifically TF-IDF based approaches, identify and include key sentences from the main source without rewriting them. This approach is particularly useful for producing short summaries that capture the main ideas of the original document. On the other hand, abstractive summarization, used in models like IndicBART, produces summaries by restructuring sentences and using phrases that may not have been mentioned in the source text. This technique has great potential to produce more coherent and natural sounding summaries, similar to those written by humans. To check the effectiveness, ROUGE scores efficiency was checked by comparing the produced and referred summaries. The results showed some good scores, with ROUGE-1 F1 of about 0.551466 on validation datasets. These results show that the models are able to learn the essential information from the input text. Nevertheless, the challenge of summarizing Hindi text remains a difficulty due to the complexity of the language and the scarcity of large-scale, high-quality annotated corpora. The study also stresses the significance of strong pre-processing strategies and more data to enhance the model accuracy and performance. Despite the recent developments in the area of text summarization for Hindi and other Indian languages, further R and D is still necessary to solve the existing problems and enhance the quality and coherence of the summarizes. The implementation of new nlp tools and construction of large, quality corpora will be critical in the evolution of this area, so that summarization models will be more accurate and applicable in real-life tasks.

Received: July 18, 2025

Volume 38 No. 1s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

**References**

[1] L. Pushpakar, A. D'souza, A. Bhalikha D. S., S. Suresh, and Nikhila G., "A brief study on Hindi text summarization using natural language processing," *Int. Res. J. Mod. Eng. Technol. Sci.*, vol. 4, no. 6, pp. 444, Jun. 2022. Accessed on: June 16, 2024. [Online].

[2] G. C. Megharaj and V. Jituri, "TFIDF model based text summerization," *Int. J. Eng. Res. Technol.*, vol. 10, no. 12, pp. 74–76, Sep. 2022. Accessed on: June 16, 2024. [Online].

[3] V. Dalal and L. Malik, "Automatic summarization for Hindi text documents using bio-inspired computing," *Int. J. Adv. Res. Comput. Commun. Eng.*, vol. 6, no. 4, pp. 686–687, Apr. 2017. Accessed on: June 16, 2024. [Online].

[4] M. Supreet, K. Goel, and M. Gupta, "Automatic Hindi text summarization using selection and elimination approach," *Int. J. Eng. Appl. Sci. Technol.*, vol. 5, no. 4, pp. 259–266, Aug. 2020. Accessed on: June 18, 2024. [Online].

[5] A. Agarwal, S. Naik, and S. Sonawane, "Abstractive text summarization for Hindi language using IndicBART," in *Proc. Forum Inf. Retrieval Eval. (FIRE)*, vol. **3395**, Dec. 2022, pp. 409–417. Accessed on: June 18, 2024. [Online].

[6] N. Desai and P. Shah, "Automatic text summarization using supervised machine learning technique for Hindi language," *Int. J. Res. Eng. Technol.*, vol. 5, no. 6, pp. 363–364, Jun. 2016. Accessed on: June 19, 2024. [Online].

[7] J. L. Neto, A. A. Freitas, and C. A. A. Kaestner, "Automatic text summarization using a machine learning approach," *Proc. 16th Brazilian Symp. Artif. Intell. (SBIA)*, **vol. 2507**, pp. 205–215, Nov. 2002. Accessed on: : June 21, 2024. [Online].

[8] A. Rajasekaran and R. Varalakshmi, "Review on automatic text summarization," *Int. J. Eng. Technol.*, vol. 7, no. 2.33, pp. 456–460, Jun. 2018. Accessed on: June 19, 2024. [Online].

[9] L. Peng, X. Ma, and Z. Teng, "Detection of stopwords in classical Chinese poetry," *Int. J. Adv. Comput. Sci. Appl.*, vol. 16, no. 2, pp. 258–263, Feb. 2025. Accessed on: June 19, 2024. [Online].

[10] D. R. Faria, A. I. Weinberg, and P. P. Ayrosa, "Multimodal affective communication analysis: Fusing speech emotion and text sentiment using machine learning," *Appl. Sci.*, vol. 14, no. 15, pp. 6631, Aug. 2024. Accessed on: June 21, 2024. [Online].

[11] A. Wibisono, "Music album review rating prediction using transformers," M.S. thesis, Leiden Inst. Adv. Comput. Sci., Leiden Univ., Leiden, The Netherlands, 2023. Accessed on: June 23, 2024. [Online].

[12] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, Oct. 2018. Accessed on: June 23, 2024. [Online].

[13] R. Boorugu and G. Ramesh, "A survey on NLP based text summarization for summarizing product reviews," in *Proc. 2nd Int. Conf. Inventive Res. Comput. Appl. (ICIRCA)*, Coimbatore, India, Jul. 2020, pp. 352–356. Accessed on: June 23, 2024. [Online].

Received: July 18, 2025

Volume 38 No. 1s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

[14] P. Janjanam and C. P. Reddy, "Text summarization: an essential study," in *Proc. Int. Conf. Comput. Intell. Data Sci. (ICCIDS)*, Feb. 2019, pp. 1–6. Accessed on: June 23, 2024. [Online].

[15] R. Dedhia, H. P. Pachgade, A. P. Malani, N. Raul, and M. Naik, "Study on abstractive text summarization techniques," in *Proc. Int. Conf. Emerg. Trends Inf. Technol. Eng. (ic-ETITE)*, Feb. 2020, pp. 1–8. Accessed on: June 26, 2024. [Online].

[16] N. S. Shirwandkar and S. Kulkarni, "Extractive text summarization using deep learning," in Proc. 4th Int. Conf. Comput. Commun. Control Autom. (ICCUBEA), Aug. 2018, pp. 1–5. Accessed on: June 26, 2024. [Online].

Received: July 18, 2025