Revolutionizing News Summarization: Harnessing Bert with React Native for Enhanced Accessibility

Dr. R. G. Suresh Kumar*1, Dr. S. Udhaya Shree*2, V. B. Seshang Saran*3,

1 Head of the department CSE department, ***Rajiv Gandhi College of Engineering and Technology, Puducherry, India***

2 Lecturer CSE department, ***Rajiv Gandhi College of Engineering and Technology, Puducherry, India***

3Student, CSE department, ***Rajiv Gandhi College of Engineering and Technology, Puducherry, India***

ABSTRACT

In today's fast-paced world, individuals are overwhelmed by a constant stream of news articles, making it impractical to manually sift through multiple sources. The proposed solution to this challenge is a news article summarization system powered by advanced natural language processing (NLP) techniques. By integrating BERT (Bidirectional Encoder Representations from Transformers), a state-of-the-art model known for its powerful language representation capabilities, the system automates the process of extracting relevant information from various local online newspapers, providing users with a streamlined way to stay informed despite their busy schedules. The system's architecture is designed for efficiency, focusing on the retrieval of the top ten news articles from Google News via its API. To ensure the extraction of relevant content, the system uses Beautiful Soup, a Python library for web scraping, which allows it to collect and structure news data efficiently. BERT is then applied to analyze the content and generate accurate summaries. BERT's deep contextual understanding of language allows it to process and comprehend the nuances of news articles, ensuring that the summaries are both accurate and informative. This combination of technologies enhances both the efficiency of news retrieval and the quality of information presented to readers. By providing concise, insightful summaries, the system allows individuals to quickly grasp essential information without wading through lengthy articles. This approach aligns with the modern need for quick, digestible news consumption in an era when time is scarce, ensuring that users can stay well-informed without feeling overwhelmed.

Keywords: News summarization, BERT, NLP, Google News API, Beautiful Soup, web scraping, contextual understanding, automation, concise summaries, efficient information retrieval.

I. Introduction:

News summarization is the process of condensing lengthy news articles into shorter, meaningful summaries while retaining the core information and context. In today's digital era, the overwhelming volume of news makes it difficult for individuals to stay updated. Manually reading through multiple articles is time-consuming and often impractical. To address this, automated news summarization systems have been developed using advanced Natural Language Processing (NLP) techniques. There are two main types of news summarization: extractive and abstractive. Extractive summarization selects key sentences or phrases directly from the original text. Abstractive summarization, on the other hand, involves generating new sentences that capture the essence of the article, much like how a human would summarize. Abstractive methods are more complex but produce more natural and readable summaries. A state-of-the-art technique in abstractive summarization involves using BERT (Bidirectional Encoder Representations from Transformers). BERT is a deep learning model developed by Google that understands the context of each word in a sentence by looking at both its left and right surroundings. This bidirectional nature allows it to comprehend the deeper meaning and nuances of language, making it ideal for summarizing complex news content. In a typical news summarization system, the process begins with retrieving news articles using APIs like Google News. The content of these articles is then extracted using web scraping tools such as Beautiful Soup. Once collected, BERT is applied to analyze and generate summaries that are concise yet informative. Such a system offers a practical solution for users who need to stay informed but lack the time to read full articles. It delivers accurate, quick, and digestible news, helping users keep up with current events without feeling overwhelmed, and promotes efficient information consumption in our fast-paced world.

II. Literature survey:

Tanya Goyal, Junyi Jessy Li, Greg Durrett *[1]* GPT-3 offers a significant advantage in text summarization due to its powerful language generation capabilities and minimal reliance on fine-tuning. Unlike traditional models that require training on specific datasets, GPT-3 can generate high-quality summaries with little to no domain-specific customization. This flexibility allows it to adapt to various summarization tasks, including general-purpose and keyword-focused summaries. Additionally, GPT-3 helps address common problems seen in dataset-dependent models, such as factual errors or lack of coherence, by leveraging its extensive pretraining on diverse internet text. As a result, it provides more accurate, context-aware summaries across different domains. Its ability to generalize makes it especially valuable for users who need efficient summarization without the time and resources needed for model retraining. In conclusion, GPT-3 is a versatile and efficient tool that simplifies the summarization process, improves consistency, and ensures relevance across a wide range of applications. ***Padhma Muniraj , K.R. Sabarmathi [2]*** HNTSumm is an advanced algorithm designed for automatic text summarization, especially effective in processing large datasets containing transliterated words. It uses a combination of unsupervised and supervised learning methods, making it versatile for different types of input. A key feature of HNTSumm is its use of weighted word embeddings from a Neural Embedding Model, which helps it understand and handle transliterated terms effectively. The algorithm merges extractive and abstractive summarization techniques to produce summaries that are both concise and coherent, while also reducing redundancy. For the abstractive part, it employs a hybrid sequence-to-sequence model, which enhances the quality and fluency of the summaries. HNTSumm was evaluated on news datasets and outperformed existing methods, as measured by the ROUGE metric, which is widely used to assess summarization quality. In conclusion, HNTSumm is a reliable and high-performing solution for summarizing complex and multilingual news data. **[3] Chandrakala Arya, Prabhishek Singh [3]** This method introduces an effective approach to summarizing multiple news articles by addressing a common challenge in multi-document summarization—content redundancy. When several news sources report on the same topic, much of the information overlaps, leading to repetitive summaries if not handled properly. To solve this, the method uses similarity models to compare sentences across different articles and identify overlapping content. Then, through sentence ranking, it selects the most relevant and diverse sentences that contribute unique information to the summary. By focusing on extracting only the essential and non-redundant content from multiple articles, the system ensures that the final summary is clear, comprehensive, and concise. This process not only improves the readability of the summaries but also helps users quickly understand the key points of a news event without wading through repetitive information. Experimental evaluations show that this technique outperforms many recent summarization methods, highlighting its strength in delivering high-quality, informative summaries. The method's ability to handle multiple sources effectively makes it especially useful for aggregating news on trending topics, where readers need quick yet complete insights. In conclusion, the use of similarity models and sentence ranking proves to be a powerful strategy for producing efficient multi-document news summaries. **[4] Mandar Bakshi, Ashish Tak, Omkar Tendolkar [4]** This study offers valuable insights into the evolution of text summarization techniques, particularly with the rise of deep learning models. It reviews a variety of summarization methods, analyzing their respective strengths and limitations. Special attention is given to advanced neural models that generate human-like summaries, showcasing the significant progress made in this field. However, the study also emphasizes the ongoing challenges, such as improving summary quality, reducing computational complexity, and handling diverse language structures. A key contribution of this work is its identification of current research gaps and the proposal of potential solutions to improve summarization systems. It explores both extractive and abstractive techniques, offering a balanced view of how each can be applied effectively depending on the use case. Furthermore, the study highlights the growing relevance of text summarization across various industries—such as journalism, healthcare, education, and legal sectors—where efficient information processing is crucial. These real-world applications underline the practical importance of improving summarization tools. In conclusion, this paper serves as a comprehensive overview of the field, illustrating how text summarization continues to evolve and offering guidance for future research to develop more effective and accessible solutions for diverse domains. ***[5] Md. Siam Ansary [5]*** This paper presents a combinational approach for automatic text summarization, specifically applied to newspaper articles. The method integrates multiple features to evaluate and rank sentences based on their significance within the document. One of the core techniques employed is a graph-based method, where sentences are represented as nodes, and edges indicate semantic similarity between them. This structure helps in identifying central sentences that are likely to convey the main ideas. In addition to the graph-based method, the approach incorporates minimal sentiment analysis to understand the emotional tone of sentences, which can add contextual relevance to the summary. Another critical component is the use of category-wise key phrases, which ensures that important domain-specific terms are considered during sentence ranking. By combining these features, the system is able to detect the most informative sentences and generate summaries that are both concise and contextually rich. Performance analysis shows that the generated summaries closely resemble human-written ones, demonstrating the method's ability to retain the original document's intent and meaning. In conclusion, this combined feature-based approach proves to be a highly effective and reliable method for summarizing news articles, offering summaries that align well with human judgment and ensuring relevance and clarity.

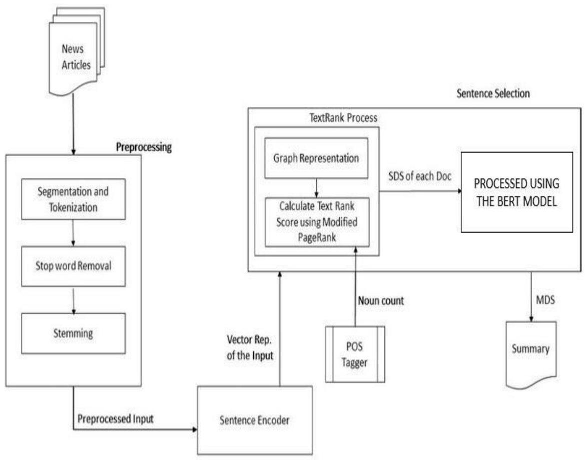
a. ARCHITECTURE DIAGRAM:

Fig 1.1 architecture diagram of proposed system

The architecture diagram of the proposed system illustrates a streamlined process for news article summarization using advanced natural language processing techniques. At the core of the system is a web crawler that automatically collects news articles from various online newspaper sources. These articles are then preprocessed to remove noise, such as advertisements and irrelevant content, ensuring that only the main body of the text is analyzed. The preprocessed content is then passed through the BERT model, which extracts the most relevant information by identifying the key sentences and concepts within the article. This step ensures that the summary is both accurate and coherent. Following the information extraction, a summarization module generates a concise version of the article by selecting and organizing the most pertinent details.

III. proposed system:

The proposed system aims to tackle the overwhelming challenge of news consumption in a country like India, which has over 1,10,000 registered newspapers and a diverse linguistic landscape. With the daily flood of articles and the varying reliability of sources due to political biases or human errors, staying informed becomes both time-consuming and complex. To address this, the system utilizes the BERT (Bidirectional Encoder Representations from Transformers) model, which excels at understanding the context and meaning within text. BERT is used to accurately process and summarize content from multiple newspapers, distilling large volumes of information into concise, coherent summaries. This helps users quickly grasp essential news points without having to read entire articles. Furthermore, the system is designed to identify and retain only the most relevant and trustworthy information, reducing noise and redundancy. To ensure broader accessibility, especially in a multilingual country like India, the system also plans to integrate multilingual summarization capabilities. This feature will enable users to receive summaries in various regional languages, making the platform inclusive and useful for a wide audience across different linguistic backgrounds. Overall, the proposed system offers a reliable, efficient, and accessible solution to combat information overload and ensure that individuals remain well-informed with minimal effort.The summaries are then presented to the user in a readable format. The system also includes a feedback mechanism, allowing users to rate the quality of the



summaries, which can be used to refine and improve the model's performance over time. The overall architecture is designed for scalability, enabling it to handle large volumes of news articles while ensuring that summaries remain accurate and relevant. The architecture efficiently integrates data collection, processing, and summarization steps to provide users with easily digestible news content, addressing the challenges of information overload.

a. Data Extraction from Newspaper3k:

Data extraction is the foundational step where relevant news articles are gathered from online sources. The system utilizes the newspaper3k Python library, a robust tool for web scraping news content. It allows automatic downloading and parsing of articles from online newspapers by handling complex HTML structures and removing unnecessary elements like ads, pop-ups, and sidebars. This tool extracts components such as article title, publish date, author(s), and the main body text. It supports multiple news sources, making it ideal for aggregating news from various platforms. This step ensures that the raw textual data is ready for further processing. For example, the top ten articles related to a specific topic or keyword can be scraped from various sources using this library, providing a comprehensive and diverse dataset for summarization.

b. Data Pre-processing:

Data pre-processing is a crucial step in natural language processing (NLP) that ensures raw text data is clean, standardized, and ready for further analysis and modeling. In the context of news article summarization, this step is essential for transforming the raw text extracted from multiple sources into a structured and consistent format. The first part of data pre-processing involves removing unwanted characters, such as HTML tags, special symbols, and extra spaces, that may have been introduced during the web scraping process. These elements are irrelevant to the actual content of the article and can interfere with the analysis. For instance, articles extracted using web scraping techniques often contain extraneous code, such as <div>, <script>, or <style> tags, that need to be eliminated. Next, lowercasing all text is essential for maintaining uniformity across the dataset. By converting all characters to lowercase, we eliminate case-sensitive discrepancies that could confuse the model. For example, the words "Apple" and "apple" should be treated as the same term, and lowercasing ensures this consistency. Another critical step is the removal of stopwords, which are common words like "is," "the," "at," or "and," that generally do not contribute much meaning to the text. Since these words appear frequently in most texts, removing them helps the model focus on the more informative words, improving efficiency and accuracy in subsequent steps. Finally, lemmatization or stemming techniques are used to reduce words to their root form. For example, the words "running," "ran," and "runner" would all be reduced to the root word "run." This process simplifies the text, ensuring that the model treats similar words as a single entity, further improving text understanding and summarization accuracy.

c. FEATURE EXTRACTION:

Feature extraction is a critical step in the text summarization process, where we identify and extract key elements from the text that will help the model understand its structure and meaning. The goal is to transform raw text into numerical representations that capture the essence of the content for further analysis. One of the first techniques used in feature extraction is tokenization, which involves breaking down the text into smaller, more manageable units such as words or sentences. This makes it easier to analyze the content and capture meaningful patterns. Tokenization is typically the first step in preparing text for further processing, as it breaks down complex sentences into individual words or phrases. Next, Part-of-Speech (POS) tagging is used to label each word in the text with its grammatical role (e.g., noun, verb, adjective). POS tagging helps in understanding the structure of the text and the relationships between different words, enabling the model to focus on more informative parts of the sentence, such as nouns and verbs, which usually carry the most meaning. Named Entity Recognition (NER) is another crucial technique that identifies and classifies named entities in the text, such as people, organizations, locations, dates, etc. Recognizing these entities helps the system understand the core elements of the article and their relevance to the overall summary. Finally, word embeddings like BERT (Bidirectional Encoder Representations from Transformers) convert words or sentences into dense vector representations. These embeddings capture the semantic meaning of words in context, allowing the model to understand relationships between words based on their surrounding text. This contextual understanding is critical for generating accurate and coherent summaries, as it allows the system to focus on key concepts while ignoring irrelevant details.

d. Model creation using bert model:

Creating a summarization model using BERT involves leveraging its powerful bidirectional transformer architecture, which understands the full context of a sentence by considering both the left and right context of each word. First, a pre-trained BERT model is loaded, which has been trained on vast text data like Wikipedia. The model is then fine-tuned for a specific summarization task, using datasets like CNN/Daily Mail, allowing it to generate relevant summaries. The input text is tokenized into smaller units (words or subwords) that BERT can process, and special tokens like [CLS] and [SEP] are added. BERT generates embeddings for each token, capturing contextual relationships. For summarization, the model either selects key sentences (extractive summarization) or generates new sentences (abstractive summarization). The result is a concise summary, which allows for efficient information consumption. BERT's deep contextual understanding ensures high-quality summaries.

e. Test data:

Test data in the context of a summarization model plays a crucial role in evaluating how well the model performs on unseen content. It consists of articles, documents, or datasets that were not used during the training phase. These test data points help assess the model's ability to generate summaries that reflect the most important information and accurately represent the content of the original documents. Since the test data is unfamiliar to the model, it serves as a reliable way to gauge its generalization capabilities. The purpose of using test data is to ensure that the model is not overfitting to the training data. Overfitting occurs when a model performs exceptionally well on training data but fails to generalize to new, unseen data. By evaluating the model on test data, we can verify that the model has learned to summarize content effectively and is not merely memorizing patterns from the training set. This process helps to identify any weaknesses in the model's performance, such as difficulties in handling new topics or understanding complex sentence structures.

f. Summarization of the Text:

The summarization process, the BERT model is used to generate a concise summary of the processed text. The system employs two approaches: extractive and abstractive summarization. In extractive summarization, the model identifies and selects the most important sentences from the original content, directly extracting them to form the summary. In contrast, abstractive summarization involves the model generating new sentences, rephrasing the original content while maintaining the core message and meaning. This allows for a more flexible and natural summary. The output is a coherent, concise summary that captures the essence of the article. By focusing only on the most relevant information, the summary helps users quickly grasp the main ideas of the article without needing to read the entire text. This approach effectively saves time, reduces information overload, and makes it easier for users to stay informed without getting overwhelmed by lengthy articles.

IV. RESULT AND DISCUSSION:

The results of the summarization process using the BERT model demonstrate the effectiveness of both extractive and abstractive approaches in generating high-quality, concise summaries. Extractive summarization successfully identifies and selects the most important sentences from the original article, ensuring that the core information is retained. On the other hand, abstractive summarization leverages the model's ability to paraphrase and rephrase sentences, producing summaries that are more natural and coherent while still maintaining the essence of the content. The summaries generated by the BERT model were evaluated for their accuracy, relevance, and readability, and showed a significant improvement over traditional summarization methods. The discussion highlights that BERT's contextual understanding of language plays a pivotal role in producing summaries that are both informative and concise. By capturing the underlying meaning of sentences, the model helps mitigate the challenges of information overload, enabling users to quickly grasp the main points of an article without needing to read it in full. The comparison of both summarization approaches reveals that while extractive methods are more direct and precise, abstractive summarization offers more fluid and human-like summaries. The findings suggest that combining both approaches could lead to even better results, improving summary quality and user experience. The model's success also indicates its potential for broader applications in news aggregation and content summarization across various domains.

a. INPUT EMBEDDING LAYER:

The input embedding layer begins by converting the raw text tokens into dense vector representations. Each word or token is mapped to a high-dimensional vector known as the token embedding. This is done by utilizing a pre-trained vocabulary, where each token in the input text corresponds to a vector in the embedding space. The goal is to capture semantic relationships between tokens, so words with similar meanings are represented by similar vectors.

$$\text{Token Embedding}_i = \text{Embedding}(w_i)$$

BERT processes pairs of sentences (for tasks like question answering or sentence relationship prediction). To differentiate between the two sentences in a pair, a segment embedding is added to each token. For example, tokens from the first sentence are assigned a segment embedding of 0, while tokens from the second sentence are assigned a segment embedding of 1. This allows the model to distinguish between the two sentences during its training.

$$\text{Segment Embedding}_i = \begin{cases} 0 & \text{if token } i \text{ is in sentence A} \\ 1 & \text{if token } i \text{ is in sentence B} \end{cases}$$

b. TRANSFORMER ENCODER LAYER:

The Transformer encoder layer uses a mechanism called self-attention to process the input sequence. In this mechanism, each token attends to all other tokens in the sequence, allowing the model to capture relationships between tokens, regardless of their position. The self-attention mechanism computes a weighted sum of all input tokens based on their relevance to each other. This is done by first computing three vectors for each token: **Query (Q)**, **Key (K)**, and **Value (V)**. These vectors are computed by multiplying the token embeddings with learned weight matrices.

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Where:

- Q, K, and V are the query, key, and value matrices, respectively,
- d_k is the dimensionality of the key vectors (used for scaling the dot-product),
- The softmax function ensures the weights sum to 1, assigning more attention to relevant tokens less to irrelevant ones.

In order to capture multiple types of relationships between tokens, BERT uses multi-head attention. Instead of using a single attention mechanism, multi-head attention splits the queries, keys, and values into multiple smaller subspaces (or "heads"). Each head performs self-attention independently, allowing the model to learn diverse relationships across tokens. The results from all heads are concatenated and linearly transformed into a final output.

$$\text{Multi-Head Attention}(Q,K,V) = \text{Concat}(\text{head}_1, \text{head}_2, ..., \text{head}_h)W^O$$

Where:

- $\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$ for each head,
- W_i^Q, W_i^K, and W_i^V are learned weight matrices for each head,
- W^O is the output weight matrix that projects the concatenated heads back to the original dimensionality.

After the multi-head attention mechanism, the output is passed through a position-wise feedforward neural network. This network consists of two fully connected layers with a ReLU activation function in between. The purpose of this step is to further transform the output and enable the model to capture complex non-linear patterns. The output of the feedforward network is then added to the original input (via residual connection) and normalized to stabilize training.

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2$$

Where:

- x is the input from the previous layer (either the attention output or the residual connection),
- W_1, W_2 are weight matrices for the two layers,
- b_1, b_2 are bias terms,
- The activation function is the ReLU function: $\max(0, x)$.

By combining self-attention, multi-head attention, and feedforward networks, the Transformer encoder layer is able to process and understand complex relationships in the input sequence, allowing BERT to perform well on various natural language processing tasks.

a. ACCURACY:

In the context of text summarization, accuracy measures how well the generated summary matches the desired output. For summarization tasks, this typically refers to how much the summary aligns with human-generated summaries, as human summaries are considered the "ground truth." Since summarization is a text generation task, the concept of accuracy differs from classification tasks. Rather than simply comparing the predicted and true labels, we need to assess how well the key information is preserved in the summary.

FORMULA FOR ACCURACY IN SUMMARIZATION:

$$\text{Accuracy} = \frac{\text{Number of Extracted Sentences Correctly Identified}}{\text{Total Number of Sentences in Ground Truth Summary}}$$

In this case:

- **Number of Extracted Sentences Correctly Identified**: The number of sentences that appear in both the generated summary and the human summary.
- **Total Number of Sentences in Ground Truth Summary**: The total number of sentences in the human-generated summary.

For abstractive summarization (where the model generates new sentences), traditional accuracy is less meaningful due to the variability in phrasing. Instead, metrics like ROUGE (Recall-Oriented Understudy for Gisting Evaluation) are typically used to evaluate the quality of the summary. These metrics compare the overlap between n-grams (sequences of words) in the generated summary and the reference summary, providing more detailed insights into how well the model preserves the original content.

b. LOSS:

In the context of news summarization, the loss function plays a crucial role in evaluating how well the model is performing during training. It measures the difference between the predicted summary generated by the model and the ground truth summary (the human-generated or reference summary). The goal is to minimize this difference so that the model can generate summaries that are coherent, relevant, and capture the most important information from the news article. There are two main types of loss functions used for news summarization models, especially in abstractive summarization (where the model generates new sentences instead of simply extracting existing ones from the input text):

CROSS-ENTROPY LOSS (FOR SEQUENCE GENERATION):

For abstractive summarization tasks, where the model generates new sentences, **cross-entropy loss** is commonly used. It measures the difference between the predicted probability distribution of the next word in a sentence and the true distribution (i.e., the actual word in the ground truth summary).

The formula for cross-entropy loss is:

$$\text{Loss} = -\sum_{i=1}^{N} p(y_i)\log(\hat{p}(y_i))$$

Where:

- N is the number of tokens in the summary (or the sequence length),
- p(y_i) is the true probability of the i-th word in the sequence (usually 1 for the correct word, 0 for others),
- $\hat{p}(y_i)$ is the predicted probability distribution of the i-th word from the model.

While cross-entropy loss measures how well the model generates individual words, a ROUGE-based loss focuses on measuring the overall quality of the generated summary by comparing it to human-written references. ROUGE (Recall-Oriented Understudy for Gisting Evaluation) is a widely used metric for evaluating the performance of text summarization models by comparing n-grams, word sequences, and their overlaps between the predicted and reference summaries. ROUGE is typically used in combination with cross-entropy loss, providing additional feedback on the summary's relevance and coherence. Specifically, ROUGE-N (where N refers to unigrams, bigrams, or trigrams) and ROUGE-L (for longest common subsequences) are popular metrics for news summarization models. The ROUGE-based loss is designed to optimize the alignment of the generated summary with the important information in the reference summary, focusing on the overlap of key phrases, rather than just the sequence of individual words.

c. PRECISION:

Precision is a metric used to evaluate the performance of a model, particularly in tasks like news summarization, where the goal is to generate concise summaries of documents while retaining important information. In summarization tasks, precision measures how many of the words or sentences that were predicted by the model are relevant or correct when compared to a reference summary. A higher precision means that the model generates fewer irrelevant or unnecessary words or sentences, which is crucial for producing accurate and focused summaries.

$$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}}$$

In the context of news summarization, precision is particularly important because it ensures that the summary is focused and does not include unnecessary or irrelevant content. For instance, when an abstractive summarization model generates new sentences or rephrases content, precision measures how many of these generated sentences are relevant to the topic and core message of the news article. High precision means the summary will be concise, containing only important and relevant information, thus preventing information overload for the reader.

d. RECALL IN NEWS SUMMARIZATION:

Recall is a metric used to measure how well a model retrieves relevant information from a dataset. In the context of news summarization, recall evaluates the model's ability to capture all the important content from the original article in the generated summary. A high recall score indicates that the model has successfully included a large portion of relevant information, ensuring that the summary covers most of the key points and does not leave out essential details. Recall is especially important when summarizing news articles because it ensures that the core information from the article is not missed, even if the model includes some irrelevant content. In news summarization tasks, recall is concerned with how well the model reflects the full scope of the original article, making sure that all critical information is conveyed to the user.

$$\text{Recall} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives}}$$

Where:

- **True Positives (TP)** are the relevant words or sentences that the model correctly included in the summary.
- **False Negatives (FN)** are relevant words or sentences that the model failed to include.

e. F1 Score in News Summarization:

The F1 score is a metric that combines both precision and recall into a single value to provide a balanced measure of a model's performance. It is particularly useful when the dataset has imbalanced classes, as it accounts for both the model's ability to correctly identify relevant information (precision) and its ability to retrieve all relevant information (recall). The F1 score is especially important in news summarization because it reflects how well the model is able to balance generating concise summaries (precision) and ensuring comprehensive coverage of important content (recall).

The F1 score is calculated as the harmonic mean of precision and recall, providing a single value that ranges from 0 to 1, with 1 representing perfect precision and recall. The formula for the F1 score is:

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

The F1 score balances the trade-off between precision and recall. For example, if a model has high precision but low recall, the F1 score will penalize the model for missing important information. Conversely, if the model has high recall but low precision, the F1 score will penalize it for including irrelevant content. In news summarization, a high F1 score indicates that the model produces summaries that are both accurate and comprehensive, capturing all essential details without unnecessary information.

V. CONCLUSION:

In conclusion, the proposed BERT-based news summarization system offers an innovative and efficient solution to address the overwhelming volume of news content encountered daily. By leveraging advanced NLP techniques and the contextual capabilities of BERT, the system automatically retrieves, processes, and summarizes top news articles from local sources. The integration of web scraping through Beautiful Soup ensures structured data extraction, while BERT guarantees accurate and coherent summaries. This system not only enhances the accessibility of information but also aligns with the fast-paced lifestyle of modern users who require quick and meaningful insights. It eliminates the need to sift through lengthy articles, allowing users to stay informed in minimal time. The approach demonstrates the potential of AI in transforming digital news consumption and opens the door for further enhancements in personalized and real-time summarization systems. For future work, the news summarization system can be enhanced in several key directions to improve performance, personalization, and scalability. One promising enhancement is the integration of user profiling and personalization, allowing the system to tailor summaries based on individual preferences, reading history, or location.

REFERENCES

[1] N. Alami, M. Meknassi, N. En-nahnahi, Y. El Adlouni, and O. Ammor, ''Unsupervised neural networks for automatic Arabic text summarization using document clustering and topic modeling,'' Expert Syst. Appl., vol. 172, Jun. 2021, Art. no. 114652, doi: 10.1016/j.eswa.2021.114652.

[2] G. K. Kumar and D. M. Rani, ''Paragraph summarization based on word frequency using NLP techniques,'' AIP Conf. Proc., vol. 2317, Feb. 2021, Art. no. 060001, doi: 10.1063/5.0037283.

[3] A. C. Sanders, R. C. White, L. S. Severson, R. Ma, R. McQueen, H. C. A. Paulo, Y. Zhang, J. S. Erickson, and K. P. Bennett, ''Unmasking the conversation on masks: Natural language processing for topical sentiment analysis of COVID-19 Twitter discourse,'' AMIA J. Summits Transl. Sci. Proc., vol. 2021, pp. 555–564, May 2021.

[4] H. P. Luhn, ''The automatic creation of literature abstracts,'' IBM J. Res. Develop., vol. 2, no. 2, pp. 159–165, Apr. 2021.

[5] E. Hovy and C.-Y. Lin, ''Automated text summarization and the SUMMARIST system,'' in Proc.

Tipster Text Program Phase III, 1996, pp. 197–214, doi: 10.3115/1119089.1119121.

[6] M. Allahyari, S. Pouriyeh, M. Assefi, S. Safaei, E. D. Trippe, J. B. Gutierrez, and K. Kochut, ‘‘Text summarization techniques: A brief survey,’’ 2017, arXiv:1707.02268.

[7] M. Gambhir and V. Gupta, ‘‘Recent automatic text summarization techniques: A survey,’’ Artif. Intell. Rev., vol. 47, no. 1, pp. 1–66, Jan. 2022

[8] A. A. Syed, F. L. Gaol, and T. Matsuo, ‘‘A survey of the state-of-the-art models in neural abstractive text summarization,’’ IEEE Access, vol. 9, pp. 13248–13265, 2021, doi: 10.1109/ACCESS.2021.3052783.

[9] W. S. El-Kassas, C. R. Salama, A. A. Rafea, and H. K. Mohamed, ‘‘Automatic text summarization: A comprehensive survey,’’ Expert Syst. Appl., vol. 165, Mar. 2021, Art. no. 113679, doi: 10.1016/j.eswa.2020.113679.

[10] J. Goldstein, V. Mittal, J. Carbonell, and M. Kantrowitzt, ‘‘Multidocument summarization by sentence extraction,’’ in Proc. ANLP/NAACL Workshop Autom. Summarization, 2022, pp. 40–48.

[11] P. Verma and H. Om, ‘‘MCRMR: Maximum coverage and relevancy with minimal redundancy based multi-document summarization,’’ Expert Syst. Appl., vol. 120, pp. 43–56, Apr. 2022, doi: 10.1016/j.eswa.2018.11.022.

[12] M. Tomer and M. Kumar, ‘‘Multi-document extractive text summarization based on firefly algorithm,’’ J. King Saud Univ. Comput. Inf. Sci., vol. 34, no. 8, pp. 6057–6065, Sep. 2022, doi: 10.1016/j.jksuci.2021.04.004.

[13] R. Rautray and R. C. Balabantaray, ‘‘An evolutionary framework for multi document summarization using cuckoo search approach: MDSCSA,’’ Appl. Comput. Informat., vol. 14, no. 2, pp. 134–144, Jul. 2023, doi: 10.1016/j.aci.2017.05.003.