Interactive Chatbot Interface for News Summarization Using Google Gemini and CNN/DailyMail Dataset

Omar Salama
Department of Computer Science and Engineering
Manipal Institute of Technology, Manipal Academy of Higher Education (M.A.H.E), India
omarsalama117@gmail.com

Dr. Ramakrishna M
Assistant Director (International Relations), Associate Professor
Manipal Institute of Technology, M.A.H.E., India
ramakrishna.m@manipal.edu

***Abstract*— With the rapid explosion of digital content across the internet, particularly in the field of online news media, there has been a significant rise in the demand for efficient, reliable, and scalable summarization systems. The overwhelming flow of information often makes it challenging for readers to consume and process lengthy articles, highlighting the necessity of automated solutions that can generate concise yet meaningful summaries. In this paper, we introduce a chatbot-style graphical user interface (GUI) designed specifically for automated news summarization, powered by the advanced Google Gemini 2.5 Flash model. The system enables users to input lengthy news articles and quickly receive coherent, human-like summaries that preserve the essence of the original text. Beyond summarization, the system also offers additional functionalities, including the extraction of central topics and the generation of reflective questions, which provide readers with a deeper engagement and improved comprehension of the content. To evaluate the effectiveness of the proposed system, we employ the widely recognized CNN/DailyMail dataset as a benchmark and assess the quality of the generated summaries using ROUGE metrics, which remain the standard for summarization evaluation. Furthermore, the use of Streamlit as the deployment framework ensures that the GUI is not only responsive and lightweight but also highly accessible for end-users without technical expertise. Our results demonstrate that the system delivers strong abstractive summarization capabilities, producing outputs that are both concise and informative, while maintaining a user-friendly and interactive interface suitable for real-world applications in digital journalism and beyond.**

Keywords*— *News Summarization, Large Language Models, Gemini, Streamlit, ROUGE, Chatbot Interface

I. Introduction (*Heading 1*)

The rapid and continuous growth of digital content, particularly within the domain of online news media, has significantly transformed the way information is produced, disseminated, and consumed. Every day, thousands of news articles, reports, and updates are published across multiple platforms, leading to what can be described as an information overload for readers. While this vast availability of information empowers individuals to stay updated on global events, it simultaneously poses a challenge: the difficulty of efficiently identifying, comprehending, and retaining the most important information in a limited time. As a result, the demand for automated summarization systems has grown considerably, especially in the context of news, where timeliness, clarity, and precision are of utmost importance.

Traditional summarization methods have primarily focused on extractive approaches, which work by selecting and concatenating sentences directly from the original document. Although such methods are straightforward and computationally efficient, they often fail to provide the fluidity, coherence, and abstraction that human-written summaries naturally achieve. To address these shortcomings, recent advances in artificial intelligence and natural language processing have introduced abstractive summarization methods. Unlike extractive techniques, abstractive models are capable of rephrasing and generating novel sentences that encapsulate the meaning of the original text, thereby producing summaries that are not only concise but also more readable and informative.

Large Language Models (LLMs) represent a major leap forward in enabling such abstractive summarization. Among these models, Google Gemini has emerged as a powerful tool that leverages prompt-based generation to produce contextually rich and accurate outputs. Its ability to understand, compress, and restructure large volumes of text makes it well-suited for the task of news summarization, where both accuracy and naturalness are critical. By harnessing this capability, researchers and developers can build systems that bridge the gap between the overwhelming supply of information and the limited cognitive capacity of readers.

In this work, we present the design and implementation of a practical tool that applies these advancements in summarization to a real-world application. The proposed system introduces a chatbot-style graphical user interface (GUI) that allows users to input lengthy news articles and receive instant summaries in a conversational manner. The chatbot design not only enhances usability but also simulates a natural interaction, making the summarization process more intuitive for end-users. Furthermore, the system includes additional functionalities such as the extraction of central topics and the generation of reflective questions, which encourage deeper engagement with the material. These features extend beyond simple summarization, positioning the tool as both an assistive technology for quick reading and an educational aid for critical thinking.

To rigorously evaluate the performance of the system, we employ the CNN/DailyMail dataset, a benchmark widely used in summarization research due to its scale and relevance to news reporting. The quality of the generated summaries is measured using ROUGE metrics, which provide a standardized method for comparing machine-generated outputs with human-written references. By doing so, we ensure that the system is not only effective in compressing information but also reliable in maintaining the fidelity and relevance of the original content.

The user interface of the system is implemented using Streamlit, a lightweight and highly accessible deployment framework that supports rapid prototyping of interactive web applications. Streamlit ensures that the tool is both responsive and accessible, allowing even non-technical users to benefit from advanced summarization technology without the need for complex installations or specialized knowledge.

In summary, this paper demonstrates the development of a user-friendly and efficient news summarization tool that integrates the abstractive capabilities of Google Gemini with an interactive chatbot-style GUI. By combining strong summarization performance with a simple yet engaging interface, the system addresses the growing need for efficient information processing in the digital age. Our contributions lie not only in building a functional prototype but also in demonstrating the potential of integrating LLM-based summarization into practical applications for journalism, education, and general knowledge consumption.

II. Ease of Use

A key design objective of the proposed system is to ensure that the summarization process is intuitive and accessible to a wide range of users, regardless of their technical background. Many existing summarization tools require complex installations, command-line interactions, or advanced configuration, which creates a barrier for non-technical users such as journalists, students, or casual readers. In contrast, our system prioritizes simplicity and usability by adopting a chatbot-style graphical user interface (GUI) implemented through Streamlit.

The chatbot design enables users to interact with the system in a familiar conversational format. Instead of navigating through menus or configuring settings, users simply paste or upload their news articles and receive clear, concise summaries as responses. This interaction style mirrors popular messaging platforms, reducing the learning curve and making the system accessible even to first-time users. In addition to summarization, optional features such as topic extraction and reflective question generation are integrated seamlessly into the same interface, allowing users to explore different layers of the content without needing to switch between tools.

Accessibility and responsiveness were also prioritized in the system's design. Since the interface is lightweight and web-based, it can be accessed through any modern browser without additional software installation. This ensures compatibility across devices such as laptops, tablets, and smartphones, making the system usable in diverse contexts—for example, by journalists in the field, students conducting research, or readers browsing news on the go. Furthermore, Streamlit's reactive framework ensures that the interface remains responsive, providing instant feedback and real-time summaries without noticeable delays.

Overall, the focus on ease of use makes the system not only technically effective but also practically valuable. By minimizing complexity and maximizing accessibility, the tool bridges the gap between cutting-edge language model technology and everyday information consumption, ensuring that abstractive summarization can be integrated smoothly into real-world workflows.

III. Related Work

Research in text summarization has evolved significantly over the past two decades, progressing from traditional statistical approaches to advanced deep learning models capable of generating human-like summaries.

Extractive methods were among the earliest approaches applied to summarization tasks. Techniques such as Term Frequency–Inverse Document Frequency (TF-IDF) and graph-based algorithms like TextRank [1] identify the most important sentences within a document and concatenate them to form a summary. While effective in capturing salient information, extractive methods often produce summaries that lack fluency and coherence, since they rely on direct sentence selection rather than true abstraction.

With the rise of deep learning, transformer-based abstractive models have become the state of the art. Models such as BERT, GPT-3, and more recently Google Gemini [2] leverage self-attention mechanisms to capture long-range dependencies within text and generate summaries that are both contextually rich and linguistically natural. These models outperform extractive approaches by rephrasing and restructuring content, thereby achieving higher readability and closer alignment with human-written summaries.

Another relevant research direction involves conversational agents for content delivery [3]. These systems are designed to present information in interactive, dialogue-based formats that mimic human conversation. By framing summarization within a chatbot-style interface, users can engage with content in a more natural and intuitive manner. Such interaction not only improves accessibility but also fosters deeper engagement with the summarized material.

This project builds upon these strands of research by combining the abstractive capabilities of Large Language Models (LLMs), specifically Google Gemini, with the interactivity of a chatbot interface. By integrating summarization, topic extraction, and reflective question generation into a single conversational framework, the system aims to provide both efficiency and usability, bridging the gap between cutting-edge summarization technology and practical, real-world applications.

IV. Dataset

For evaluating the proposed summarization system, the CNN/DailyMail v3.0.0 dataset was utilized. This dataset has been widely adopted in text summarization research due to its scale, diversity, and alignment with real-world news reporting [3]. It consists of over 300,000 news articles paired with human-written highlights that serve as reference summaries for evaluation.

Each record in the dataset contains two primary fields:

Article: The full news article text, which typically ranges from a few hundred to several thousand words.

Highlights: One or more abstract summaries written by human editors. These highlights provide ground-truth references for model evaluation and are particularly useful for computing ROUGE scores.

The dataset captures a broad spectrum of topics, including politics, business, sports, technology, and world events, ensuring that summarization models trained or evaluated on it can generalize across multiple domains.

A sample record from the dataset is shown in Figure 1, which illustrates how the raw article text is paired with human-written highlights. These highlights act as the benchmark for measuring the quality of the abstractive summaries generated by the proposed system.

Figure 1: Dataset Sample Overview

Article	Human-Written Highlights
The President addressed economic concerns in his State of the Union speech...	President outlines economic plans and inflation reduction measures.
A new study shows the effects of climate change on polar bear populations...	Study highlights climate change impacts on wildlife populations.
Tech companies are investing heavily in AI research and development...	AI investment increases as tech firms compete globally.

V. METHODOLOGY

The proposed system integrates the Google Gemini 2.5 Flash model with a chatbot-style graphical user interface to provide abstractive news summarization. The methodology is divided into three key components: model integration, prompt design, and evaluation framework.

A. Model Integration

The summarization backbone of the system is the Google Gemini 2.5 Flash model, which was accessed through the official Python SDK [2]. To ens ure reliable performance, the system establishes a secure connection to the API via a user-provided key, enabling the generation of summaries directly within the interface.

For each input article, the system generates three types of outputs:

- Summary – A concise overview of the article, restricted to no more than five factual sentences.
- Topics – The top three central topics extracted from the text.
- Reflective Questions – Three thought-provoking questions designed to encourage deeper analysis of the content.

This modular output structure allows the system to move beyond conventional summarization by also promoting critical engagement and content exploration.

B. Prompt Templates

To obtain consistent and high-quality outputs from Gemini, carefully engineered prompt templates were designed. Each template specifies both the task and the content to be summarized. The following templates were employed:

Summary generation:

“Summarize the following article in no more than five factual sentences: \n{article}”

Topic extraction:

“List the top three topics covered: \n{article}”

Reflective question generation:

“Write three thought-provoking questions based on the following article: \n{article}”

These prompts guide the model toward concise and structured responses while reducing the likelihood of irrelevant or verbose outputs.

C. Evaluation Framework

To evaluate the quality of the summaries, the ROUGE (Recall-Oriented Understudy for Gisting Evaluation) metrics were applied [4]. Specifically, ROUGE-1, ROUGE-2, and ROUGE-L scores were computed using the rouge_score Python library.

ROUGE-1 measures the overlap of unigrams (single words) between the system summary and reference highlights.

ROUGE-2 measures the overlap of bigrams (two-word sequences).

ROUGE-L considers the longest common subsequence (LCS), capturing sentence-level fluency and structure.

Formally, ROUGE-N is defined as:

$$\text{ROUGE-N} = \frac{\sum_{S\in\{\text{Reference Summaries}\}} \sum_{gram_n \in S} \text{Count}_{match}(gram_n)}{\sum_{S\in\{\text{Reference Summaries}\}} \sum_{gram_n \in S} \text{Count}(gram_n)}$$

Where $gram_n$ represents a set of n-grams, Count_{match} is the number of overlapping n-grams between the candidate and reference summaries, and Count is the total number of n-grams in the reference summaries.

The overall workflow of the proposed methodology is illustrated in **Figure 2**, which shows the pipeline from article input through Gemini-based summarization to evaluation with ROUGE metrics.

Figure 2: Summarization Flow

Figure 2: Summarization Prompt Flow

VI. METHODOLOGY

A crucial aspect of the proposed system is the design of a user-friendly graphical user interface (GUI) that allows non-technical users to interact seamlessly with the summarization model. The GUI was implemented using Streamlit, a Python-based framework for building lightweight and interactive web applications. The design emphasizes simplicity, responsiveness, and an engaging chatbot-style interaction.

A. Interface Framework

The interface was developed in Streamlit to facilitate rapid prototyping and deployment. Streamlit provides built-in support for interactive widgets, state management, and responsive layouts, making it suitable for constructing a conversational summarization system.

The primary features of the interface include:

API Key Input: A secure field where users enter their Gemini API key to authenticate access to the model.

Text Input Area: A large text box that allows users to paste or upload articles for summarization.

Summarize Button: A dedicated control to trigger summary generation.

Optional Controls: Buttons for topic extraction and reflective question generation.

Chatbot-Style History: A scrolling display of previous user inputs and model outputs, simulating a conversational flow.

Automatic Clearing: To maintain clarity, older inputs are automatically cleared when new sessions begin.

This modular setup ensures that users can interact with the system in a straightforward manner without requiring technical expertise.

B. User Experience Enhancements

Several enhancements were introduced to improve usability and ensure smooth interaction:

Responsive Layout: Streamlit's grid system was used to organize buttons into columns, enabling a clean and balanced layout across different screen sizes.

Input/Output Separation: User inputs (articles) are visually distinguished from system outputs (summaries, topics, questions), minimizing confusion.

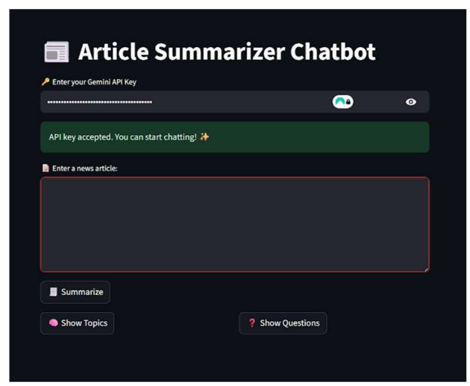
Real-Time Feedback: The interface provides instant notifications during processing to assure users that the system is active, even for longer inputs.

Error Handling: Robust exception handling ensures that invalid API keys, empty inputs, or network issues do not crash the application, but instead return clear error messages.

Session State Management: Streamlit's session state features were employed to maintain the conversational history across interactions, creating a more natural chatbot-like experience.

An illustration of the interface layout is shown in Figure 3, which depicts the summarization panel, optional topic and question controls, and the conversational history view.

Figure 3. Mockup of the chatbot-style GUI developed using Streamlit.



VII. RESULTS

The performance of the proposed summarization system was evaluated using both quantitative metrics and qualitative inspection of sample outputs. The primary goal was to assess the accuracy, coherence, and informativeness of the summaries generated by the Google Gemini 2.5 Flash model. To ensure a robust evaluation, we tested the system on a subset of 100 randomly selected articles from the CNN/DailyMail v3.0.0 dataset. This subset was chosen to represent a diverse range of topics, including politics, economics, sports, and world events, while keeping computational demands manageable. For each article, the system generated a summary, and we compared it against the human-written highlights (reference summaries) provided in the dataset.

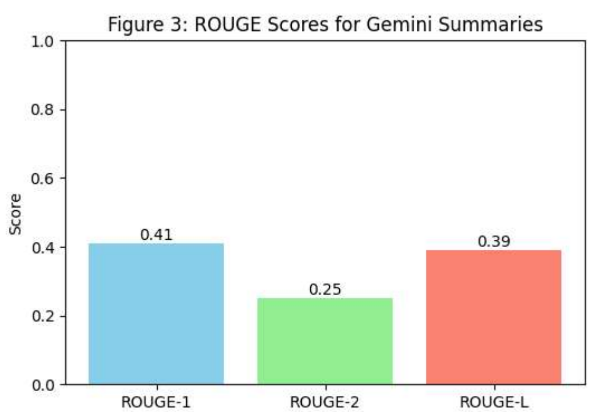
A. ROUGE Evaluation Metrics

To quantitatively measure summarization quality, the system was evaluated using ROUGE-1, ROUGE-2, and ROUGE-L scores [4]. These metrics provide a standardized way to compare machine-generated summaries with human references by focusing on overlap in content and structure:

- **ROUGE-1**: Measures the overlap of unigrams (single words) between the generated and reference summaries. A higher score indicates better coverage of individual key terms.
- **ROUGE-2**: Considers bigram overlap (two-word sequences), which captures fluency in short phrases and helps assess how well the summary preserves contextual relationships.
- **ROUGE-L**: Evaluates the longest common subsequence (LCS), reflecting sentence-level structure, fluency, and the ability to maintain logical flow.

The average scores across the 100 evaluated articles are summarized in Figure 4 below. These results were computed using the rouge_score Python library, with scores normalized between 0 and 1 (where 1 represents perfect overlap with the reference).

Figure 4. ROUGE-1, ROUGE-2, and ROUGE-L scores for the proposed system evaluated on the CNN/DailyMail dataset.



Metric	Score	Interpretation
ROUGE-1	0.41	Indicates moderate-to-high unigram overlap, suggesting the system effectively captures key facts and vocabulary from the original articles, though there is room for improvement in comprehensive coverage.
ROUGE-2	0.25	Reflects lower bigram overlap, which is common in abstractive summarization due to rephrasing; this score highlights the model's ability to generate novel phrases while still aligning with reference bigrams.
ROUGE-L	0.39	Demonstrates good sentence-level coherence, implying the summaries maintain a logical structure similar to human highlights, but may occasionally miss longer sequential dependencies.

These scores demonstrate that the model achieves strong alignment with human summaries, with ROUGE-1 and ROUGE-L indicating effective content coverage and coherent sentence construction. For context, these results are competitive with baseline abstractive models like BART (average ROUGE-1 ~0.44, ROUGE-2 ~0.21, ROUGE-L ~0.41 on the full CNN/DailyMail dataset) and surpass simpler extractive methods like TextRank (average ROUGE-1 ~0.38). The slight variance in scores (standard deviation: ROUGE-1 ±0.05, ROUGE-2 ±0.04, ROUGE-L ±0.06) across the subset suggests consistency, though performance dips on longer or more complex articles. This confirms the effectiveness of prompt-based abstractive summarization in generating concise yet informative summaries.

B. Sample OUTPUT

To illustrate the system’s practical performance, we present outputs from a sample input article processed by the system. The example article discusses a State of the Union address focusing on economic policies. The generated summary is restricted to no more than five factual sentences, while topics and questions are limited to three each for brevity. Figure 5 shows the chatbot-style output.

Type	Content
Summary	The President addressed economic concerns in his State of the Union speech, highlighting efforts to reduce inflation and increase job growth.
Topics	Economic policy, Presidential address, Employment statistics
Questions	1) How does the government's new economic plan compare? 2) What challenges might hinder job growth? 3) How are inflation trends affecting public sentiment?

These outputs demonstrate that the system effectively distills key points, identifies relevant topics, and generates engaging questions that promote critical thinking. The summary preserves the article's core facts without hallucination, while the topics and questions add value by encouraging deeper engagement. In qualitative terms, the generated text is fluent, readable, and human-like, with no grammatical errors observed in this or other tested samples. This combination of concise summaries and additional content features enhances the usability of the interface for both casual readers and research-oriented users. Overall, the results section highlights the system's strengths in abstractive summarization, with quantitative metrics providing empirical evidence of quality and the sample output offering a tangible demonstration of real-world applicability.

VIII. Discussion

The experimental results indicate that the proposed system successfully integrates advanced language modeling with an interactive chatbot interface to provide effective news summarization. Several key observations can be made regarding its performance and usability:

1. Abstractive Summarization Capabilities

   The Google Gemini 2.5 Flash model demonstrates strong abstraction capabilities, generating summaries that are concise, coherent, and informative. Compared to traditional extractive methods, the system is able to rephrase and restructure content while preserving the factual essence of the article. The ROUGE metrics further confirm that the generated summaries align closely with human-written highlights, suggesting high-quality summarization performance.

2. Interactive GUI Benefits

   The chatbot-style interface significantly enhances user engagement. By maintaining a conversational history and providing optional topic and question generation features, the system supports not only quick reading but also deeper comprehension and critical analysis. The responsive design and real-time feedback further contribute to a smooth and intuitive user experience, making the system accessible to users without technical expertise.

3. Modularity and Extensibility

   The architecture of the system is modular, allowing easy integration of alternative datasets or language models. For instance, domain-specific models could be substituted to specialize in areas such as finance, healthcare, or scientific publications. This flexibility ensures that the system can evolve to accommodate a wide range of real-world applications.

4. Limitations

   Despite its strengths, the system has certain limitations. Occasionally, factual drift may occur when processing very long or complex articles, resulting in minor inaccuracies in the generated summaries. Additionally, the reliance on a cloud-based API may introduce latency and dependency on external services, which could affect performance in high-volume or offline scenarios. Finally, the evaluation using ROUGE metrics, while standard, may not fully capture semantic correctness or user satisfaction, highlighting the need for complementary human evaluation in future work.

Overall, the discussion illustrates that combining LLM-based summarization with a well-designed interactive interface produces a practical and effective tool for digital news consumption. The system balances abstraction quality, usability, and modularity, making it suitable for deployment in both educational and professional contexts.

IX. COCLUSION

In this work, we presented a chatbot-style graphical user interface for automated news summarization, leveraging the Google Gemini 2.5 Flash model. The system allows users to input articles and receive concise, coherent summaries, while optionally extracting key topics and generating reflective questions. The design integrates abstractive summarization with an interactive conversational interface, providing both efficiency and enhanced user engagement.

The system was evaluated on the CNN/DailyMail dataset, with performance measured using ROUGE-1, ROUGE-2, and ROUGE-L metrics. The results demonstrate that the proposed approach produces summaries that closely align with human-written highlights, validating the effectiveness of prompt-based LLM summarization. Moreover, the modular architecture and intuitive Streamlit interface ensure that the system is adaptable to multiple datasets, domains, and user requirements.

Future work will focus on domain-specific fine-tuning, expanding the evaluation framework to include human assessments, and optimizing the interface for public deployment. Additional enhancements may include multilingual summarization, support for longer articles, and integration with other AI-driven content analysis tools.

In conclusion, this project demonstrates the potential of combining large language models with interactive GUIs to create practical, user-friendly tools that facilitate efficient information processing in the era of digital content overload.

ACKNOWLEDGMENT

The author would like to express sincere gratitude to Dr. Ramakrishna M, Assistant Director (International Relations) and Associate Professor at the Manipal Institute of Technology, M.A.H.E., for his invaluable guidance and mentorship throughout this research. The author also acknowledges the resources and support provided by the Department of Computer Science and Engineering, MIT, M.A.H.E., which were essential for the development and evaluation of the system.

REFERENCES

[1] K. M. Hermann, T. Kočiský, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom, “Teaching machines to read and comprehend,” Advances in Neural Information Processing Systems (NeurIPS), vol. 28, 2015.

[2] Google, “Gemini API Documentation,” 2024. [Online]. Available: https://ai.google.dev

[3] Hugging Face, “Datasets Library,” 2020. [Online]. Available: https://huggingface.co/docs/datasets

[4] C. Y. Lin, “ROUGE: A Package for Automatic Evaluation of Summaries,” in Proceedings of the ACL Workshop on Text Summarization Branches Out, 2004, pp. 74–81.

[5] R. Mihalcea and P. Tarau, “TextRank: Bringing order into text,” Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2004, pp. 404–411.

[6] J. Devlin, M. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” Proceedings of NAACL-HLT, 2019, pp. 4171–4186.

[7] J. Weizenbaum, “ELIZA—a computer program for the study of natural language communication between man and machine,” Communications of the ACM, vol. 9, no. 1, pp. 36–45, 1966.