NEWS ARTICLE TEXT SUMMARIZATION

Dr. A. Satyanarayana
Head of the department

M. Ramya Sri
B. Tech Student
Data science
Siddhartha Institute of Technology and sciences

Ch. Roopa
B. Tech Student
Data science
Siddhartha Institute of Technology and sciences

J. Gayathri
B. Tech Student
Data Science
Siddhartha Institute of Technology and sciences

E. Nishanth Reddy
B. Tech Student
Data science
Siddhartha Institute of Technology and sciences

N. Ashish Yadav
B. Tech Student
Data Science
Siddhartha Institute of Technology and sciences

Abstract: The News article summarization is a process designed to condense full-length articles into concise , informative summaries. This technique is essential in today's fast-paced world, where the volume of news content can be overwhelming Summarization methods fall into two main categories: extractive and abstractive. Extractive summarization focuses on identifying and selecting key sentences from the original text, while abstractive summarization involves generating new phrases and sentences that encapsulate the core ideas.

1. INTRODUCTION

In today's fast-paced world, staying informed is essential, but the sheer volume of news articles can be overwhelming. Text summarization offers a solution by condensing lengthy articles into concise summaries, providing readers with key information efficiently. By leveraging natural language processing and machine learning techniques, text summarization algorithms can identify the most important points of an article, making it easier for individuals to grasp the main ideas without having to read the entire piece. This technology has numerous applications, from helping professionals stay updated in their fields to enabling quick news consumption for busy individuals. In this article, we will explore the significance of text summarization in the realm of news consumption, its underlying technologies, and its potential impact on how we consume information in the digital age. Text summarization is revolutionizing the way we interact with information overload. With the exponential growth of online content, from news articles to research papers, users face a daunting task of sifting through vast amounts of text to extract relevant information. Text summarization algorithms streamline this process by automatically generating concise summaries, saving users time and effort. These algorithms employ advanced natural language processing techniques such as sentence extraction, abstraction, and semantic analysis to distill the essence of a text while preserving its key points.

1.1 DESCRIPTION

One of the primary benefits of text summarization is its ability to cater to diverse reading preferences. Whether individuals prefer skimming through headlines or delving into in-depth analysis, summarization algorithms can adapt to their needs by generating summaries of varying lengths and complexities. This flexibility enhances user engagement and accessibility, enabling individuals with limited time or attention spans to stay informed without feeling overwhelmed. Furthermore, text summarization has significant implications for professionals across various industries. In fields such as finance, healthcare, and law, where staying abreast of the latest developments is critical, summarization algorithms can provide timely insights and facilitate informed decision-making. By aggregating information from multiple sources and distilling it into concise summaries, these algorithms empower professionals to stay competitive in dynamic environments. Moreover, text summarization contributes to the democratization of information by making complex topics more accessible to a wider audience. Whether it's breaking news, scientific discoveries, or policy updates, summarization algorithms enable individuals from diverse backgrounds and expertise levels to grasp the significance of complex topics without requiring specialized knowledge.

1.2 PROBLEM STATEMENT

"Requirement to build a document summarization product to save time and efforts of people and to use human resources efficiently."

The amount of information is increasing every day. Thus finding relevant data becomes hectic and time consuming, more over not all the data is relevant to the user's topic of interest. In order to find relevant data for user's search and to save time is it necessary to have a small summary of the documents. Summary made by humans is time consuming and tedious. Thus there is a need for automatically summarizing the text document to save time and to get quick results. Automatic Summarization can be defined as the art of condensing large text documents into few lines of summary, giving important information

1.3 SCOPE AND MOTIVATION

The intention of our system, text summarization, is to express the content of a document in a condensed form that meets the needs of the user. Far more information than can realistically be digested is available on the World-Wide Web and in other electronic forms. There are many categories of information (economy, sports, health, technology) and also there are many sources (news site, blog, SNS), it is not possible to read everything one would want to read and so some form of information condensation is needed. So to make an automatically & accurate summaries feature will helps us to understand the topics and shorten the time to do it.

1.4 PROJECT OBJECTIVES

- To develop a system which will summarize a text document.
- To pre-process the text document to be analyzed by text summarization algorithm.
- To create an algorithm for extracting the most important text in the document.
- To create and train an NLP based data set for better sentence extraction. (Stop words, prefixes, suffixes, etc.)
- To define a number of text features which are used for scoring the importance of a sentence in text.
- To calculate score for each sentence of text.
- To select the best sentences for summary.
- To serve the end user with summary of text which the individual has uploaded.

2. LITERATURE REVIEW

Here we will elaborate the aspects like the literature survey of the project and what all projects are existing and been actually used in the market which the makers of this project took the inspiration from and thus decided to go ahead with the project covering with the problem statement.

2.1 LITERATURE SURVEY

1) NewsIN: A News Summarizer and Analyzer [IJRASET]: This paper proposes a news summarization system specifically designed for online news articles. It explores techniques for sentence selection and highlights the importance of tailoring the summarization process for news content.
2) Deep Learning Approaches: Some recent studies explore the application of deep learning techniques, such as recurrent neural networks (RNNs) and transformers, for text summarization. These models often achieve state-of-the-art performance by learning hierarchical representations of input text and generating concise summaries.
3) Multimodal Summarization: With the increasing availability of multimedia content, there's a growing interest in multimodal summarization, which combines information from multiple modalities such as text, images, and videos to generate comprehensive summaries. Research in this area investigates fusion strategies and the integration of visual and textual features for summarization tasks.
4) Domain-Specific Summarization: Text summarization techniques often need to be adapted to specific domains, such as biomedical literature, legal documents, or social media posts. Research in domain-specific summarization explores domain-specific features, terminology, and discourse patterns to improve the relevance and accuracy of summaries in specialized domains.

2.1.1 Extraction Based Measurements

Paper of Jason Weston et al [6] have proposed a supervised learning for deep architectures, if one jointly learns an embedding task using unlabeled data was improved. Researchers used shallow architectures already showed two ways of embedding to improve generalization. First is embedding unlabeled data as a separate preprocessing step (i.e., first layer training) and the second is used for embedding as a regularized (i.e., at the output layer). More importantly, they have generalized these approaches to the case where, have train a semi-supervised embedding jointly with a supervised deep multi-layer architecture on any (or all) layers of the network, and showed have been could bring real benefits in complex tasks..

2.1.2 Behavioral measurements

F. kyoomarsi et al [3] have presented an approach for creating text summaries. Used fuzzy logic and word-net, they have been extracted the most relevant sentences from an original document. The approach utilizes fuzzy measures and inference on the extracted textual information from the document to found the most significant sentences. Experimental results reveal that come within reach of extracted the most relevant sentences when compared to other commercially available text summarizers.

2.1.3 Matrix Based Measurement

Binwahlan et al [4] has incorporated fuzzy logic with swarm intelligence; so that risks, uncertainty, ambiguity and imprecise values of choosing the features weights (scores) could be flexibly tolerated. The weights obtained from the swarm experiments were used to adjust the text features scores and then the features scores and then the features scores were used as inputs for the fuzzy inference system to produce the final sentence score. The sentences were ranked in descending order based on their scores and then the top n sentences were selected as final summary.

2.1.4 Features Based Measurements

Kiani et al [2] proposed a novel approach that extracts sentences based on an evolutionary fuzzy inference engine. The evolutionary algorithm uses GA and GP in concert. The genetic algorithm is used to optimize the membership functions and genetic programming is used to optimize the rule sets. The problem of competing conventions in fuzzy system optimization is thereby reduced by decoupling the two major categories of optimization in fuzzy systems. Fitness function is chosen to consider both local properties and global summary properties by considering various features of a given sentence such as its relative number of used thematic words as well its location in the whole document.

2.2 EXISTING SYSTEM

2.2.1 Manual Summarization

In the existing system, articles are summarized manually by editors and journalists, following a practice deeply ingrained in the history of media organizations. This manual process involves reading through the entirety of an article and condensing its key points into a shorter version.

Drawbacks:

Despite its historical precedence, manual summarization suffers from several drawbacks. Firstly, it is a time-consuming process, as it requires individuals to read through each article thoroughly and then craft a summary. This can be particularly problematic when dealing with large volumes of articles, leading to delays in the summarization process. Additionally, manual summarization is inherently subjective, as the quality and depth of the summary can vary depending on the individual summarizer's interpretation of the content. This subjectivity introduces inconsistency and may result in summaries that do not accurately reflect the original article's key points. Moreover, human error is a significant risk in manual summarization, as summarizers may inadvertently omit crucial information or misinterpret the article's content.

Consistency Challenges:

One of the primary challenges associated with manual summarization is ensuring consistency in the quality and style of the summaries produced. Because summarization is a manual task, different summarizers may have varying levels of expertise and interpretive abilities, leading to inconsistencies in the summaries they produce. These inconsistencies can manifest in the form of differences in tone, level of detail, and overall comprehensiveness across summaries of similar articles. As a result, readers may encounter summaries that lack coherence and fail to provide a clear representation of the original article's content.

2.3 PROPOSED SYSTEM

News article text summarization where it does overcome all the drawbacks that mentioned in existing system includes:

Automatic Summarization: The system automates the process of summarizing news articles, reducing the need for manual effort and saving time for users. This automation enables the rapid processing of large volumes of news articles, making it suitable for applications requiring real-time summarization.

Customizable Summarization Parameters: Users can customize parameters such as the length of the extractive summary and the word limits for the abstractive summary. This flexibility allows users to tailor the summaries to their specific needs or preferences, enhancing the utility of the system.

Combination of Extractive and Abstractive Techniques: The system leverages both extractive and abstractive summarization techniques to provide a comprehensive summary of news articles. Extractive summarization preserves important sentences from the original article, while abstractive summarization generates summaries that capture the essence of the article in new language. This combination improves the coverage and quality of the summaries, catering to different user preferences and use cases.

Utilization of Pre-trained Models: The system utilizes pre-trained transformer-based models fine-tuned for summarization tasks, such as the T5 model. These models, trained on large-scale datasets, capture complex language patterns and semantic relationships, leading to more accurate and contextually relevant summaries.

Interactive User Interface: The system provides an interactive user interface through Streamlit, allowing users to input news articles and visualize the summaries easily. The interface includes sliders and widgets for parameter selection, enhancing user experience and usability.

3. REQUIREMENT ANALYSIS AND PLANNING

In requirements analysis encompasses those tasks that go into determining the needs or conditions to meet for a new or altered product or project, taking account of the possibly conflicting requirements of the various stakeholders, analyzing, documenting, validating and managing software or system requirements. Project planning is part which relates to the use of schedules such as Gantt charts to plan and subsequently report progress within the project environment. Initially, the project scope is defined and the appropriate methods for completing the project are determined

3.1 FUNCTIONAL REQUIREMENT

Requirement Analysis will cover the topics like the Functional, Non-Functional and the specific requirements of the project and touching all the software and the hardware requirements as well.

3.1.1 Text Summarizer Requirements

- The system should provide text parser functions which can take the whole text and separate into sentences, paragraphs and words.
- The system should provide text feature function which can take the necessary part and obtain a feature vector
- The system should provide a well-trained Autoencoder to generate better inputs for classifier.
- The system needs a classifier which is well trained to select summary sentences.
- The system should provide a sentence modifier to beautify and polish output text while changing some words with their synonyms etc.

3.1.2 Summarize Web Page Requirements

- The system should provide a “Web Sayfasını Özetle” button with complete functionality. When clicked on this button, browser extension send the html of the current web page to the server
- A function which detect body part and select text. This function needs to extract unnecessary text from html.
- The system should provide communication between server and client with necessary network functions such send and receive.

3.1.3 Summarize File Requirements

- The system should provide a “Summarize” button with complete functionality. After user selected target file, the user presses the “button and web page application send the file to the server
- A set of functions provide the reading from file depends on file extension
- The system should provide communication between server and client with necessary network functions such send file and receive file.

3.1.4 Summary Setting Requirements

- The system should take parameters such as summary length from user before summarizing.

3.1.5 Train System Requirements

- The system should provide login screen for admin.
- The system should provide taking new data from admin to train Autoencoders or classifiers to improve reliability

3.2 FUNCTIONAL REQUIREMENT

3.2.1 Usability

The system should be easy to use. The user should reach the summarized text with one button press if possible. Because one of the software’s features is timesaving. The system also should be user friendly for admins because anyone can be admin instead of programmers. Training the Autoencoders and classifiers are used too many times, so it is better to make it easy.

3.2.2 Reliability

This software will be developed with machine learning, feature engineering and deep learning techniques. So, in this step there is no certain reliable percentage that is measurable. Also, user provided data will be used to compare with result and measure reliability. With recent machine learning techniques, user gained data should be enough for reliability if enough data is obtained. The maintenance period should not be a matter because the reliable version is always run on the server which allow users to access summarization. When admins want to update, it take long as upload and update time of executable on server. The users can be reach and use program at any time, so maintenance should not be a big issue.

3.2.3 Performance

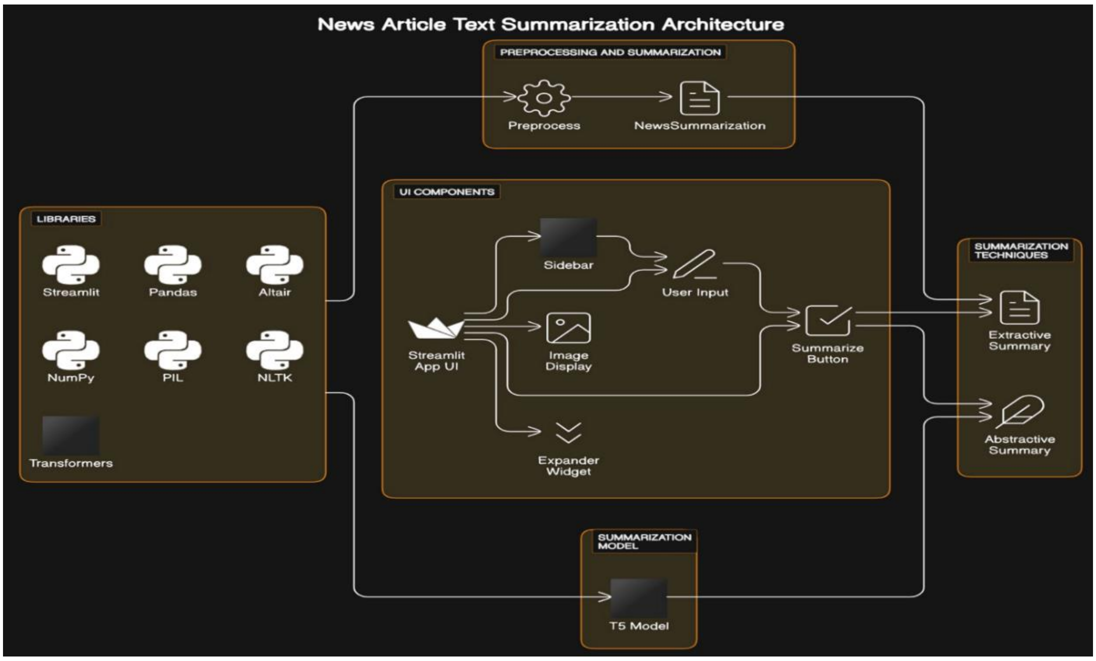
Calculation time and response time should be as little as possible, because one of the software’s features is timesaving. Whole cycle of summarizing a page/file should not be more than 30 seconds in order to 3 pages long document. The capacity of servers should be as high as possible. Calculation and response times are very low, and this comes with that there can be so many sessions at the same times. The software only used in Turkey, than do not need to consider global sessions. 1 minute degradation of response time should be acceptable. The certain session limit also acceptable at early stages of development. It can be confirmed to user with “servers are not ready at this time” message.

3.2.4 Supportability

The system should require C, Java, Python and Matlab knowledge to maintenance. If any problem acquire in server side and deep learning methods, it requires code knowledge and deep learning background to solve. Client side problems should be fixed with an update and it also require code knowledge and network knowledge.

4. ARCHITECTURE

The architecture presented consists of two Python scripts: highlights.py and summarize.py, which collaborate to create a web application for news article summarization.



4.1 highlights.py:

Streamlit Interface Setup: This script leverages Streamlit, a popular Python library for creating web applications, to build the user interface. It establishes the layout, user input elements, and output presentation for the summarization application.

Model Loading: Utilizes the Hugging Face transformers library to load a pre-trained summarization model (shivaniNK8/t5-small-finetuned-cnn-news). This model is responsible for generating abstractive summaries based on input news articles.

User Interaction: Provides users with text input areas to submit news articles they wish to summarize. Offers sliders for users to adjust parameters such as summary length for extractive summarization and word limits for abstractive summarization. Includes a "Summarize!" button for triggering the summarization process based on user input.

Summarization Output: Displays the extractive summary and abstractive summary of the input news article. Renders the summarization results on the Streamlit app interface for user consumption.

4.2 summarize.py:

Text Preprocessing: Defines a Preprocess class responsible for preparing the input text data before summarization. Offers methods for tasks like converting text to lowercase, tokenizing sentences and words, removing stopwords, and lemmatizing tokens.

Model Evaluation: Implements a NewsSummarization class dedicated to summarization-related functionalities. Includes methods for extractive summarization, computing Rouge scores to evaluate the quality of summaries, and evaluating summarization models against datasets.

Word Embedding Model Training: Provides functions for training Word2Vec and GloVe word embedding models, which are essential for understanding the semantic relationships between words in the text data.

5. SYSTEM REQUIREMENT SPECIFICATION

4.3 Software Requirements

Operating System: The software should be compatible with commonly used os such as windows, Linux and Mac.

Python: The code is written in Python programming language, so Python runtime environment needs to be installed on the system. Python version 3.7 or later is recommended.

Python Libraries: Install the required Python libraries using pip or conda package managers.:

- Streamlit
- pandas
- altair
- NumPy
- PIL (Python Imaging Library)
- nltk (Natural Language Toolkit)
- transformers (Hugging Face Transformers library)
- gensim
- rouge_score
- matplotlib
- wordcloud

Development Environment: A code editor or integrated development environment (IDE) such as Visual Studio Code, PyCharm, or Jupyter Notebook can be used for writing and running the code.

5.2 Hardware Requirements

Processor (CPU): A multi-core processor with decent processing power is recommended for handling text processing tasks efficiently.

Memory (RAM): At least 4GB of RAM is recommended for smooth execution, especially when working with large datasets or running Complex summarization.

Storage: Sufficient disk space to store the application code, libraries, and any generated data. This requirement can vary depending on the size of the dataset and the models used.

6. SYSTEM DESIGN

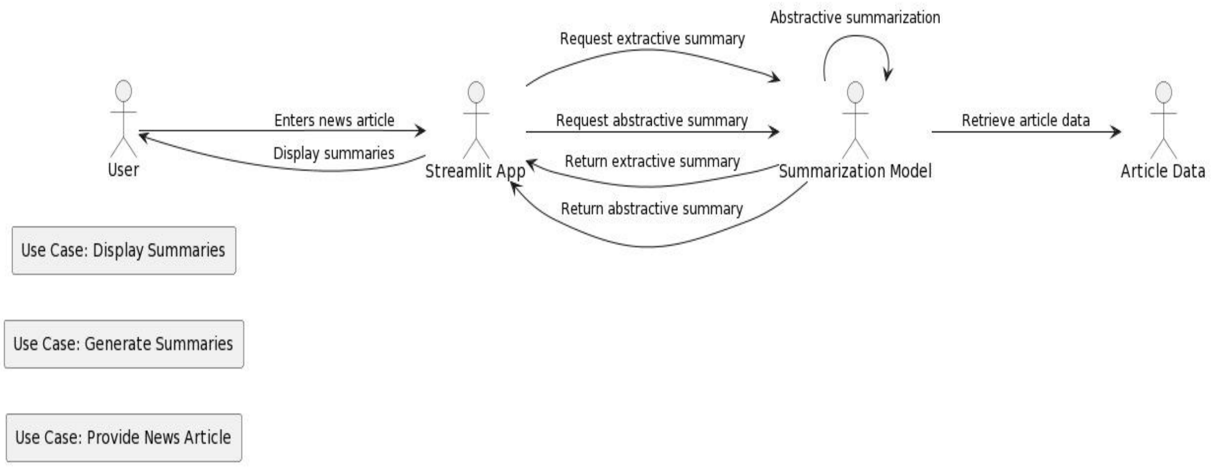
System Design refers to the process of defining the architecture, components, modules, interfaces, and data for a software system. It involves making critical decisions on how the software will meet its functional and non-functional requirements.

6.1.UML (Unified Modeling Language) Diagrams

UML (Unified Modeling Language) Diagrams are visual representations used to model software systems.

6.1.1. Use Case Diagram:

- **Purpose**: To illustrate the interactions between users (actors) and the system.
- **Components**: Use cases (functionalities), actors (users), and relationships between them.

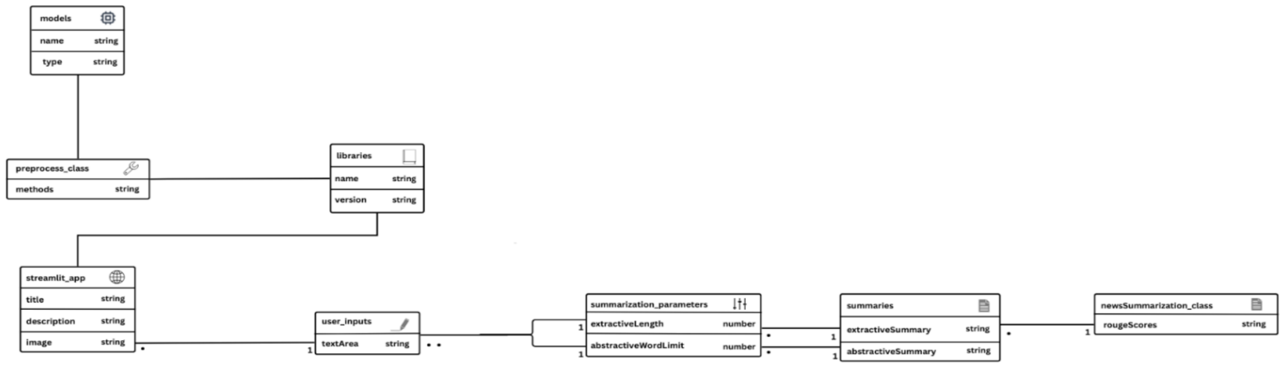

Abstractive summarization
Request extractive summary
Enters news article
Display summaries
Request abstractive summary
Return extractive summary
Return abstractive summary
Retrieve article data
User
Streamlit App
Summarization Model
Article Data
Use Case: Display Summaries
Use Case: Generate Summaries
Use Case: Provide News Article

- **Usage**: Use case diagrams show how users interact with the system, identifying key functionalities like user registration, data upload, analysis, and result export.

Fig 6.1.1 Use Case Diagram

6.1.2. ER Diagram:

- **Purpose**: To describe the structure of the software system, including classes, their attributes, methods, and relationships.
- **Components**: Classes, attributes, methods, associations, and inheritance relationships.



- **Usage**: Class diagrams provide an overview of the system's object-oriented design, representing entities like users, chat data, analysis components, and more.

Fig 6.1.2 ER diagram

6.1.3. Sequence Diagram:

- **Purpose**: To depict the interactions and message exchanges between objects (actors) in the system over time.
- **Components**: Lifelines (objects), messages, activations, and the order of message flow.
- **Usage**: Sequence diagrams illustrate how the system components interact during processes like user registration and data analysis, showing the chronological order of actions.

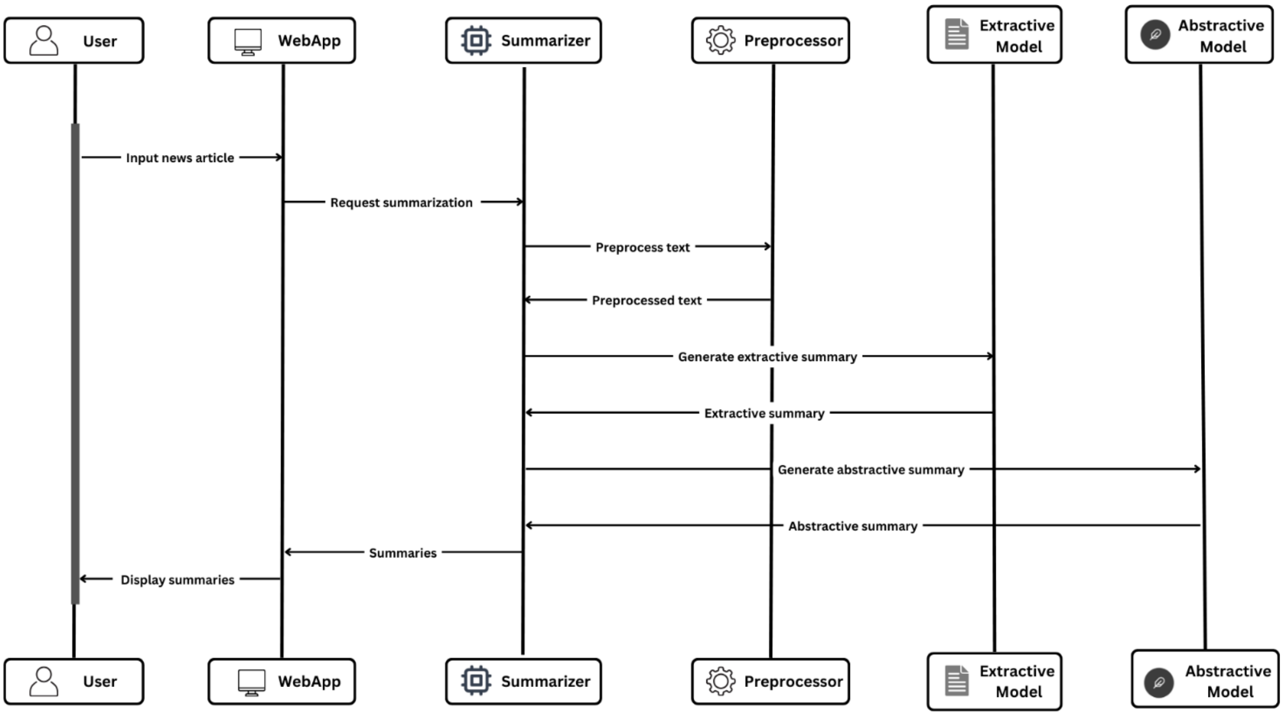


Fig 6.1.3 Sequence diagram

6.1.4. Flowchart

Flowchart is a visual representation of a process or algorithm, often using symbols and arrows to illustrate the steps, decisions, and flow of control within the process.

Purpose: Flowcharts are designed to visualize the step-by-step sequence of actions or operations within the software system. They provide a clear and easy-to-understand way of representing the logic and flow of the application's functionalities.

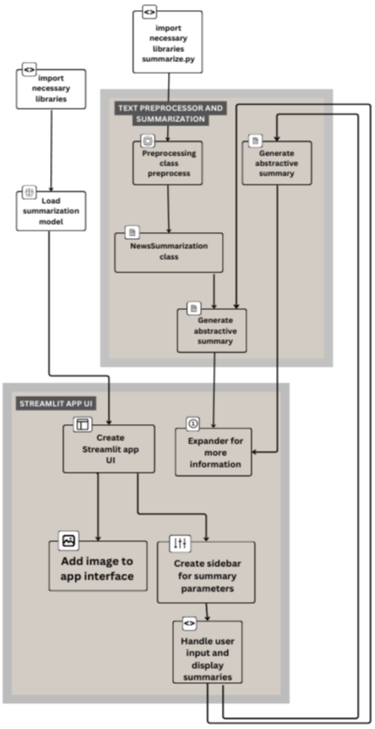
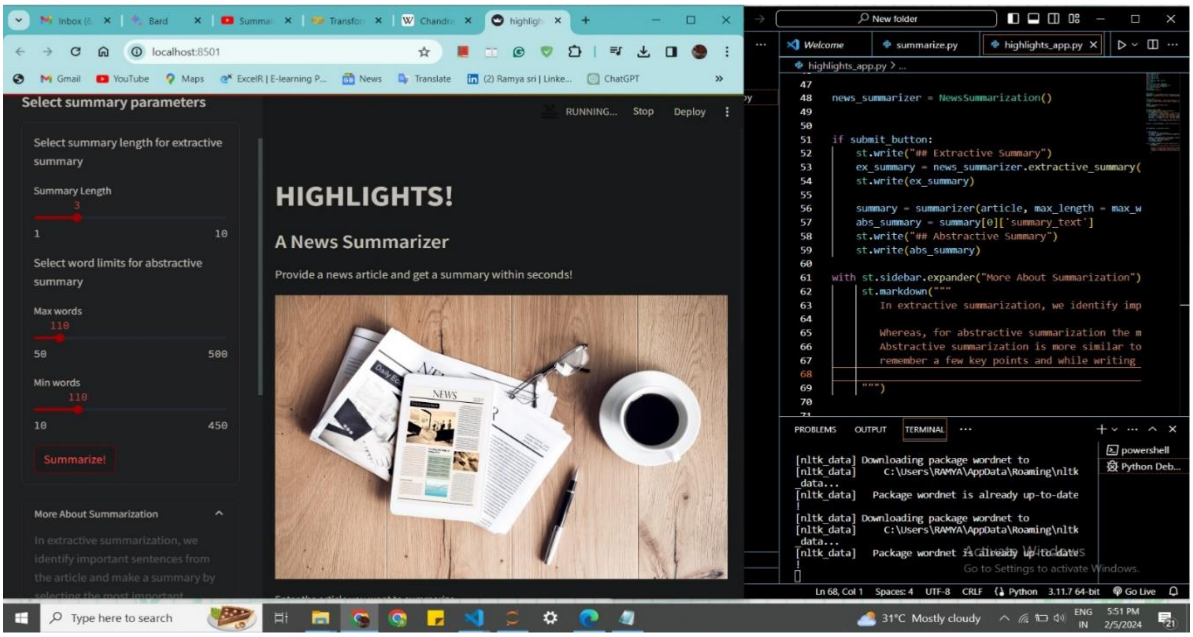

import necessary libraries summarize.py
import necessary libraries
TEXT PREPROCESSOR AND SUMMARIZATION
Preprocessing class preprocess
Generate abstractive summary
Load summarization model
NewsSummarization class
Generate abstractive summary
STREAMLIT APP UI
Create Streamlit app UI
Expander for more information
Add image to app interface
Create sidebar for summary parameters
Handle user input and display summaries

Fig 6.1.4 Flow chart

7. RESULT



Select summary parameters

Select summary length for extractive summary

Summary Length 3 (1–10)

Select word limits for abstractive summary

Max words 110 (50–500)

Min words 110 (10–450)

Summarize!

More About Summarization

In extractive summarization, we identify important sentences from the article and make a summary by selecting the most important

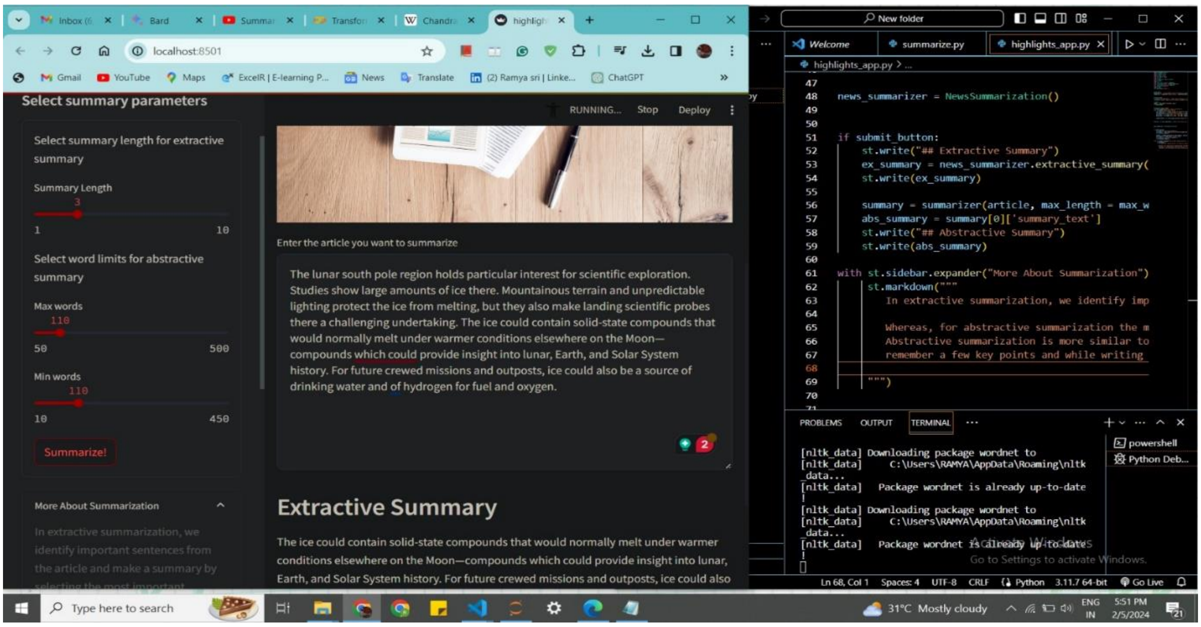
HIGHLIGHTS!

A News Summarizer

Provide a news article and get a summary within seconds!

```
news_summarizer = NewsSummarization()


if submit_button:
    st.write("## Extractive Summary")
    ex_summary = news_summarizer.extractive_summary(
    st.write(ex_summary)

    summary = summarizer(article, max_length = max_w
    abs_summary = summary[0]['summary_text']
    st.write("## Abstractive Summary")
    st.write(abs_summary)

with st.sidebar.expander("More About Summarization")
     st.markdown("""
        In extractive summarization, we identify imp

        Whereas, for abstractive summarization the m
        Abstractive summarization is more similar to
        remember a few key points and while writing

     """)
```

```
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\RAMYA\AppData\Roaming\nltk
_data...
[nltk_data]   Package wordnet is already up-to-date
!
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\RAMYA\AppData\Roaming\nltk
_data...
[nltk_data]   Package wordnet is already up-to-date
!
```



Enter the article you want to summarize

The lunar south pole region holds particular interest for scientific exploration. Studies show large amounts of ice there. Mountainous terrain and unpredictable lighting protect the ice from melting, but they also make landing scientific probes there a challenging undertaking. The ice could contain solid-state compounds that would normally melt under warmer conditions elsewhere on the Moon—compounds which could provide insight into lunar, Earth, and Solar System history. For future crewed missions and outposts, ice could also be a source of drinking water and of hydrogen for fuel and oxygen.

Extractive Summary

The ice could contain solid-state compounds that would normally melt under warmer conditions elsewhere on the Moon—compounds which could provide insight into lunar, Earth, and Solar System history. For future crewed missions and outposts, ice could also

Select summary parameters

Select summary length for extractive summary

Summary Length 3 (1–10)

Select word limits for abstractive summary

Max words 110 (50–500)

Min words 110 (10–450)

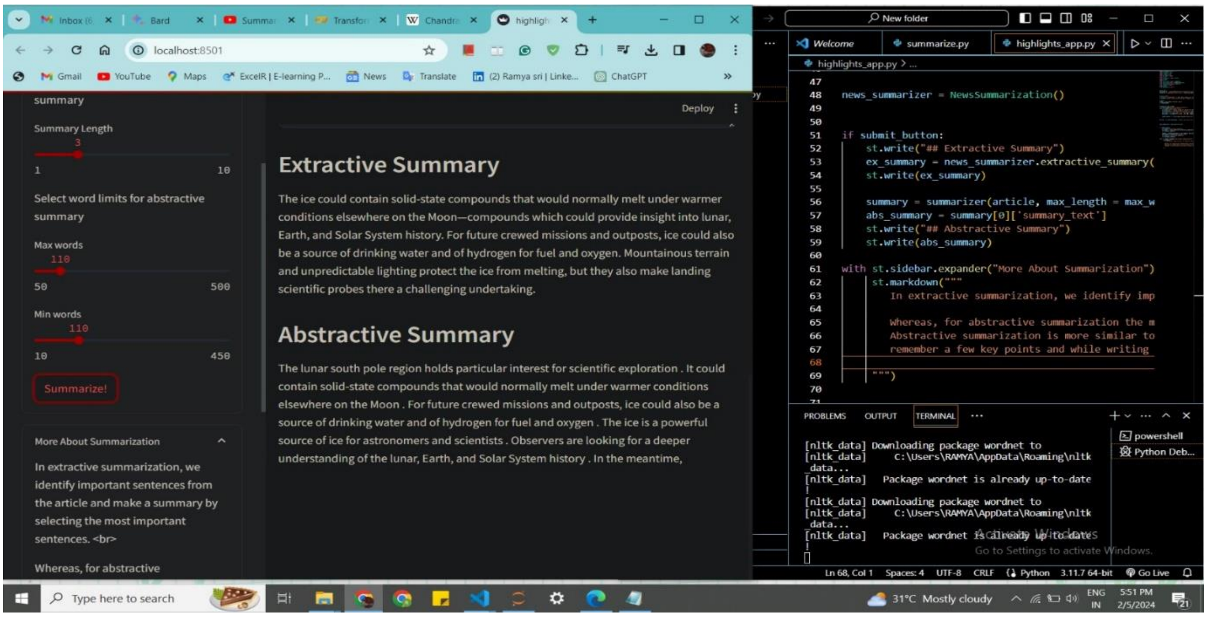
Summarize!

More About Summarization

In extractive summarization, we identify important sentences from the article and make a summary by selecting the most important

```
news_summarizer = NewsSummarization()


if submit_button:
    st.write("## Extractive Summary")
    ex_summary = news_summarizer.extractive_summary(
    st.write(ex_summary)

    summary = summarizer(article, max_length = max_w
    abs_summary = summary[0]['summary_text']
    st.write("## Abstractive Summary")
    st.write(abs_summary)

with st.sidebar.expander("More About Summarization")
     st.markdown("""
        In extractive summarization, we identify imp

        Whereas, for abstractive summarization the m
        Abstractive summarization is more similar to
        remember a few key points and while writing

     """)
```

```
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\RAMYA\AppData\Roaming\nltk
_data...
[nltk_data]   Package wordnet is already up-to-date
!
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\RAMYA\AppData\Roaming\nltk
_data...
[nltk_data]   Package wordnet is already up-to-date
!
```



Extractive Summary

The ice could contain solid-state compounds that would normally melt under warmer conditions elsewhere on the Moon—compounds which could provide insight into lunar, Earth, and Solar System history. For future crewed missions and outposts, ice could also be a source of drinking water and of hydrogen for fuel and oxygen. Mountainous terrain and unpredictable lighting protect the ice from melting, but they also make landing scientific probes there a challenging undertaking.

Abstractive Summary

The lunar south pole region holds particular interest for scientific exploration . It could contain solid-state compounds that would normally melt under warmer conditions elsewhere on the Moon . For future crewed missions and outposts, ice could also be a source of drinking water and of hydrogen for fuel and oxygen . The ice is a powerful source of ice for astronomers and scientists . Observers are looking for a deeper understanding of the lunar, Earth, and Solar System history . In the meantime,

More About Summarization

In extractive summarization, we identify important sentences from the article and make a summary by selecting the most important sentences.

Whereas, for abstractive

summary

Summary Length 3 (1–10)

Select word limits for abstractive summary

Max words 110 (50–500)

Min words 110 (10–450)

Summarize!

```
news_summarizer = NewsSummarization()


if submit_button:
    st.write("## Extractive Summary")
    ex_summary = news_summarizer.extractive_summary(
    st.write(ex_summary)

    summary = summarizer(article, max_length = max_w
    abs_summary = summary[0]['summary_text']
    st.write("## Abstractive Summary")
    st.write(abs_summary)

with st.sidebar.expander("More About Summarization")
     st.markdown("""
        In extractive summarization, we identify imp

        Whereas, for abstractive summarization the m
        Abstractive summarization is more similar to
        remember a few key points and while writing

     """)
```

```
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\RAMYA\AppData\Roaming\nltk
_data...
[nltk_data]   Package wordnet is already up-to-date
!
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\RAMYA\AppData\Roaming\nltk
_data...
[nltk_data]   Package wordnet is already up-to-date
!
```

CONCLUSION

In conclusion, the project successfully developed a news summarization system capable of generating both extractive and abstractive summaries, providing users with valuable insights into news articles in an efficient and user-friendly manner. While the system demonstrates significant achievements, ongoing refinement and optimization efforts are essential to further enhance its performance and usability. Overall, the project lays a solid foundation for future advancements in text summarization technology and its applications in various domains.

The project's development of a news summarization system that can generate both extractive and abstractive summaries reflects its versatility and adaptability. This dual capability ensures that users have access to summaries that suit their preferences and needs, whether they prefer.concise, point-based extracts or more nuanced, paraphrased content. By providing users with concise yet comprehensive summaries of news articles, the project contributes to enhancing the user experience of accessing and digesting large volumes of information.Users can quickly grasp the key points and essential insights of news articles without having to read through lengthy texts, thereby saving time and effort.

REFERENCES

1. Aik, L. E. (2008). A study of neuro-fuzzy system in approximation-based problems. Neuro-Fuzzy System ANFIS : Adaptive Neuro-Fuzzy Inference System, 24(2), 113–130.
2. Albertos, P. (1998). Fuzzy logic controllers. Methodology. Advantages and drawbacks. In X Congreaso Espanol Sobre Technologias Y Logica Fuzzy (ESTYLF), Sevilla, España, pp. 1–11.
3. Babar, S. A., & Patil, P. D. (2015). Improving performance of text summarization. In Procedia – procedia computer science (Vol. 46, pp. 354–363). Elsevier Masson SAS.
4. Dixit, R. S., & Apte, P. S. S. (2012). Improvement of text summarization using fuzzy logic based method. IOSR Journal of Computer Engineering (IOSRJCE), 5(6), 5–10.
5. Fattah, M. A., & Ren, F. (2008). Automatic text summarization. International Journal of Computer, Electrical, Automation, Control and Information Engineering, 2(1), 90–93.
6. Kontostathis, A., & Kulp, S. (2008). The effect of normalization when recall really matters. International conference on Information and Knowledge Engineering (IKE), Las Vegas, NV, pp. 96–101.
7. Kumar, Y. J., Goh, O. S., Halizah, B., Ngo, H. C., & Puspalata, C. S. (2016). A review on automatic text summarization approaches. Journal of Computer Science, 12(4), 178–190. ISSN 1549-3636.
8. Loganathan, C., & Girija, K. V. (2014). Investigations on hybrid learning in ANFIS. International Journal of Engineering Research and Applications, 2(10), 31–37.
9. Megala, S. S., Kavitha, A., & Marimuthu, A. (2014). Enriching text summarization using fuzzy logic. International Journal of Computer Science and Information Technologies, 5(1), 863–867.