IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

TEXTBYTE: A SUMMARIZATION TOOL

[1]Prof. Kirti Patil, [2]Yashashri Sonar, [3]Shahid Hannure, [4]Aarti Thorat, [5]Shraddha Ghagare

[1]Assistant Professor, [2,3,4,5] UG Students

Department of Computer Engineering,

Alard College of Engineering and Management Pune India

Abstract: In today's digital era, navigating through vast volumes of online information poses a formidable challenge. Extracting accurate insights from this abundance demands efficient tools like Textbyte. These tools play a pivotal role in condensing extensive content into concise summaries, empowering users to grasp complex data swiftly. While Wikipedia articles are renowned for their depth, abstraction techniques aid in distilling this information without diluting its essence. Abstracting content involves minimizing its volume while retaining its core meaning, a crucial aspect for streamlined research and information processing. Textbyte leverages advanced features, including YouTube video summarization, PDF chatbot integration, and file summarization. Users can input a video link for summarization, inquire about a provided PDF, or receive a summary of a text file. Notably, Textbyte employs cutting-edge NLP techniques, utilizing the LLM architecture and an extractive model, powered by SQL lite, to ensure accurate and efficient summarization.

Index Terms - Text Summarization, chat with PDF, YouTube Video Summarization, NLP, LLM Architecture, Extractive model, Abstractive model.

I. INTRODUCTION

Text Summarization Process:

The system employs natural language processing (NLP) algorithms to analyze lengthy news articles or documents. These algorithms scan the text to identify important information, key points, and relevant details. Using various techniques such as statistical methods, machine learning, or deep learning models, the system determines which parts of the text are most essential. It then generates concise summaries that capture the essence of the original content while significantly reducing its length.

Chat with PDF:

The system could feature a chat interface where users can interact with PDF documents directly. Users might ask questions about specific sections of a PDF, and the system could provide relevant answers or summaries. It could allow users to highlight portions of the PDF and discuss them in real-time with others through the chat interface. Users could also request summaries of lengthy PDF documents through the chat, and the system could provide condensed versions for easier consumption.

YouTube Summarization:

The system could analyze YouTube videos and generate brief summaries of their content. It might extract key points, topics, or timestamps from the video and present them in a summarized format. Users could request summaries of lengthy videos or select specific sections they want summarized. Summarized versions of YouTube videos could include text summaries, bullet points, or even visual representations of the main points discussed.

For this study secondary data has been collected. From the website of KSE the monthly stock prices for the sample firms are obtained from Jan 2010 to Dec 2014. And from the website of SBP the data for the macroeconomic variables are collected for the period of five years. The time series monthly data is collected on stock prices for sample firmsand relative macroeconomic variables for the period of 5 years. The data collection period is ranging from January 2010 to Dec 2014. Monthly prices of KSE -100 Index is taken from yahoo finance.

II. RELATED WORK

Since there is so much information available on the web, identifying relevant information is key to using it effectively. The content of the text is important in solving this problem. It provides a way to present long messages in context while retaining the main points. This technique not only helps you understand the text faster, but also makes it easier to extract important ideas from different sources. The explosive growth of digital content, including newspapers, magazines, social media and news, has created a huge need to find and understand quality information. Keywords are important to meet these requirements and allow users to understand the importance of information without needing to access all the information. Xindong Wu published News Recommendations and Filters in [11], which uses recommendations from Google News and World News and turns them into personalized articles based on user preferences. David Reis, Bruno Piedade, and others introduced the use of Docker and Docker Compose in Docker development and debugging in [5]. Sulochana Devi and colleagues completed the data collection in [1]. It is a technique that does not use point-by-point sentences but uses the main points of the original text. The video summarization process now requires good knowledge. Subtitled video content is the fastest way to create content as it is easier and faster to create text. Kai Jiang and Xi Lu [7] stated that natural language processing is based on many disciplines such as linguistics, computer science and mathematics, and

wisdom has now become an important research topic. Rapid advances in natural language processing have led to machine translation research. Natural language processing is an important part of artificial intelligence and it starts with machine translation. Chellalamalla Mamatha was successfully designed, implemented and tested in [10]. DevOps (Integrated Development and Operations) is a software development process that involves the integration of the roles of software developers and information technology (IT) professionals. The main goal is to deliver software products faster and reduce errors, thus increasing efficiency. K. G. Kharade and some authors developed the NLTK library for coding in [4]. As a result of this process, content that can be used without changing the meaning of the content emerges. Finally, users will be able to get more information from maps. Business analysts, business leaders, governments, students, researchers, and educators all need to expand their resources. I think we should have managers who try to process as much information as possible in less time. Saeedeh Gholamrezadeh and some authors in [13] discussed the analysis of different methods in different contexts and strategies to evaluate the writing process. M.F. Mrida et al. [3] performed an analysis of various levels of the general ATS field: simple theory, dataset analysis, feature extraction architectures, efficient summarization algorithms, models, evaluation matrices and existing architectures with previous research history are introduced and the limitations of ATS are introduced as taught. Current ATS Systems and Methods. And expose their problems; this will encourage researchers to address these limitations and solve new problems in ATS.

III. PROPOSED WORK

The proposed work of Textbyte a Summarization Tool. In this model, we implemented a summarization tool that summarizes the content or news of a news article. It also summarizes YouTube videos by providing links to the videos. PDF summaries and PDF chat are also available.

3.1. System Architecture

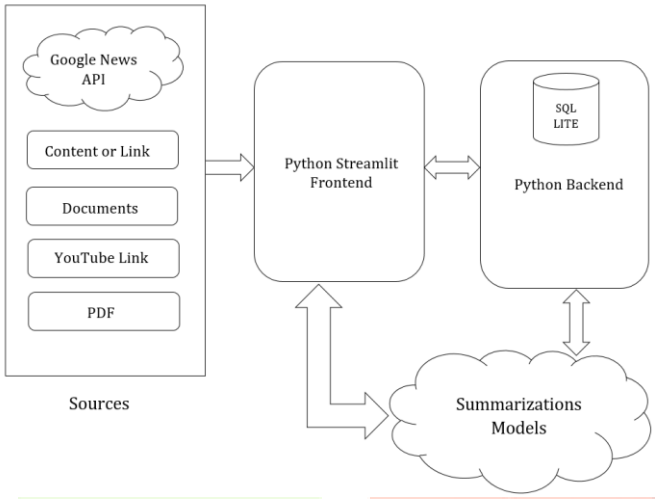

Fig. 3.1: System Architecture

The architecture of a summarization tool built using Python, Streamlit for frontend, Python for backend, and SQLite database can be designed as follows:

3.1.1. Frontend (Streamlit):

Streamlit is used to create a user-friendly web interface for the summarization tool. Streamlit allows for easy creation of interactive web applications using Python scripts. The frontend provides a user interface where users can input text/documents to be summarized and interact with the summarization tool.

3.1.2. Backend (Python):

The backend is responsible for handling user requests from the frontend and performing the summarization tasks. It contains the core logic of the summarization tool, including text processing, model execution, and summarization generation. Python's natural language processing (NLP) libraries such as NLTK, spaCy, or Transformers (using Hugging Face's transformers library) can be used for text pre-processing and summarization.

3.1.3. Summarization Models:

The backend employs machine learning or deep learning models for text summarization. Models can include extractive summarization techniques (e.g., TextRank, Gensim's summarization module) or abstractive summarization models (e.g., Transformer-based models like BERT, GPT, T5). The choice of summarization model depends on factors like the type of input data, desired summary output, and computational resources available.

3.1.4. SQLite Database:

SQLite is used as a lightweight relational database management system. It stores data such as user preferences, past summaries, or any other relevant information. The database can be utilized for caching frequently accessed data, storing user session information, or logging user activities.

3.1.5. Interaction Flow:

User interacts with the frontend interface provided by Streamlit, inputting text/documents to be summarized. The frontend sends the user's input to the backend server. The backend pre-processes the input text, selects the appropriate summarization model, and generates a summary. The summary is returned to the frontend and displayed to the user. Optionally, the summary or other relevant data can be stored or logged in the SQLite database for future reference or analysis.

3.1.6. Deployment:

The summarization tool can be deployed on a server or cloud platform such as AWS, Google Cloud Platform. Deployment considerations include scalability, performance, and security requirements.

By integrating Streamlit for frontend, Python for backend, SQLite database for storage, and various summarization models, the architecture enables the creation of a robust summarization tool with a user-friendly interface.

3.2. Algorithm

Step 1: Text Cleaning.
Step 2: Sentence Tokenization.
Step 3: Word Tokenization.
Step 4: Frequency Analysis.
Step 5: Summarization.

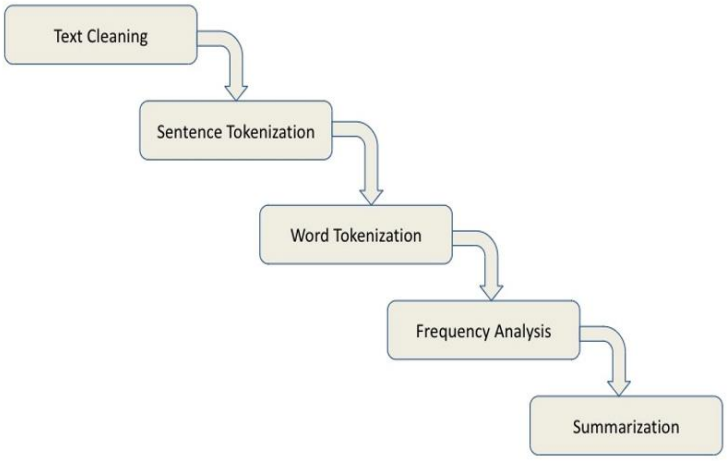

Fig. 3.2: Algorithm

3.2.1. Text Cleaning:

This step involves pre-processing the input text to remove any noise or irrelevant information that could affect the quality of the summary. Common text cleaning techniques include removing special characters, punctuation, stop words, and any other formatting inconsistencies.

3.2.2. Sentence Tokenization:

Sentence tokenization involves breaking the text into individual sentences. Each sentence is treated as a separate unit for analysis and summarization.

3.2.3. Word Tokenization:

Word tokenization involves breaking down each sentence into its constituent words or tokens. This step allows for further analysis at the word level, such as frequency analysis.

3.2.4. Frequency Analysis:

Frequency analysis involves counting the occurrence of each word in the text. Words that appear more frequently are likely to be more important in conveying the overall meaning of the text. This analysis helps in identifying the key terms or phrases that should be included in the summary.

3.2.5. Summarization:

Sentences containing the most important words or phrases are selected as topics based on frequency analysis and other criteria (such as sentence length or position). These selected sentences are usually those that contain the most famous words or have the greatest impact on the text. The final summary is created by combining the selected sentences.

3.3. LLM Architecture for PDF Chatbot Integration:

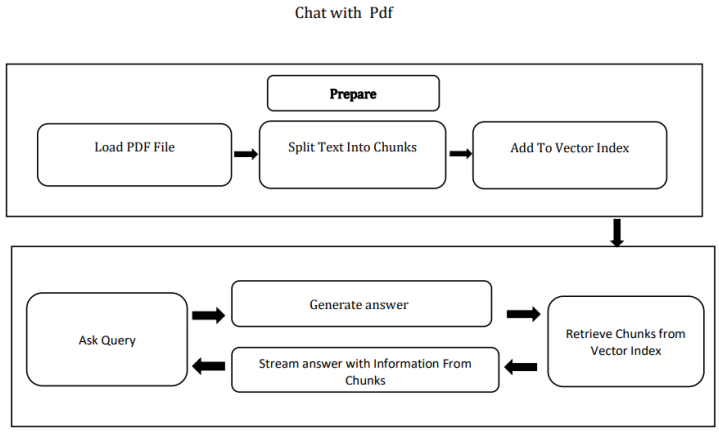

Fig.3.3: LLM Architecture

3.3.1. PDF Loading:

Implement a feature to load PDF files into the system. This should include handling various PDF formats and ensuring compatibility with different file structures.

3.3.2. Text Segmentation:

Develop algorithms to split the text content of the PDF into smaller, manageable chunks. These chunks should be of a suitable size for indexing and retrieval purposes.

3.3.3. Vector Indexing:

Create a vector index where the segmented text chunks are stored. Each chunk should be indexed appropriately for efficient retrieval.

3.3.4. Retrieval Mechanism:

Users should be able to retrieve specific text chunks from the indexed vector quickly and seamlessly. This could involve implementing search functionalities or direct access methods.

3.3.5. Streaming Introduction:

Provide a feature to stream introductory sections from the text chunks. This could involve presenting users with summaries or key points from the beginning of each chunk to give them a context before diving into detailed queries.

3.3.6. Query Interaction:

Allow users to interact with the system by inputting queries or requests for specific information. The system should be able to process these queries and retrieve relevant text chunks from the indexed vector.

By integrating these features, users can efficiently navigate through PDF documents, access relevant information, and interact with the system to extract insights or answers to their queries effectively.

3.4. Infrastructure Architecture

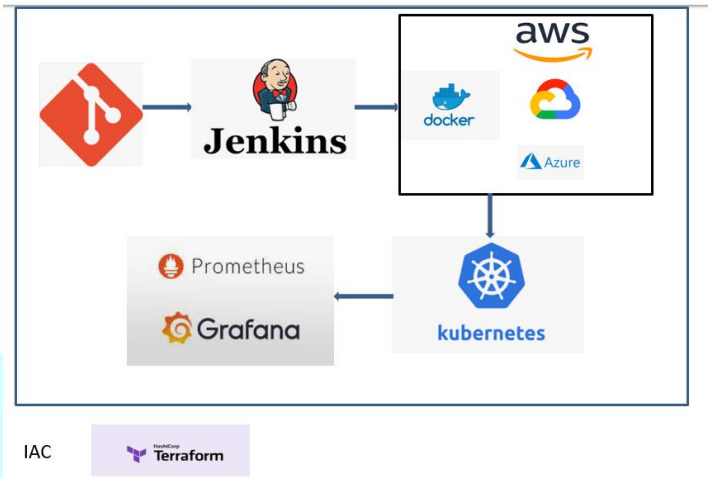

Fig.3.4: Infrastructure Architecture

3.4.1. Git - Code:

Git is used as the version control system (VCS) to manage the source code of the summarization tool and related configurations. Developers collaborate on code changes using branches and merge them into the main branch through pull requests. All changes to the code base, including infrastructure code, are versioned and tracked in Git.

3.4.2. Jenkins - Continuous Integration and Continuous Delivery:

Jenkins serves as the CI/CD server responsible for automating the integration, testing, and deployment processes. Upon each code commit to the Git repository, Jenkins triggers a series of automated tests, including unit tests, integration tests, and code quality checks. If the tests pass successfully, Jenkins generates artifacts ready for deployment and initiates the deployment process.

3.4.3. Docker - Containerization:

Docker is utilized for containerizing the summarization tool and its dependencies. The application and its runtime environment are packaged into Docker containers, ensuring consistency across different environments. Docker containers encapsulate the application, its dependencies, and configurations, making it easy to deploy and run the application on any platform that supports Docker.

3.4.4. Kubernetes - Container Orchestration:

Kubernetes is employed for container orchestration, managing the deployment, scaling, and operation of Docker containers. Kubernetes abstracts away the underlying infrastructure and provides tools for deploying, scaling, and managing containerized applications. It ensures high availability, fault tolerance, and scalability by automatically distributing containers across a cluster of nodes and handling resource allocation and scheduling.

3.4.5. Prometheus and Grafana - Continuous Monitoring:

Prometheus is used as the monitoring tool to collect and store metrics from various components of the infrastructure. Grafana is employed for visualizing and analyzing the collected metrics through customizable dashboards. Together, Prometheus and Grafana provide continuous monitoring of the summarization tool, allowing teams to monitor performance, track resource utilization, and troubleshoot issues in real-time.

3.4.6. Terraform - Infrastructure as Code:

Terraform is utilized for defining and provisioning infrastructure resources as code. Infrastructure components such as virtual machines, networks, storage, and Kubernetes clusters are defined in Terraform configuration files. Terraform automates the provisioning and management of infrastructure resources across different cloud providers or on-premises environments, ensuring consistency and reproducibility.

IV. RESULT AND OUTPUT

The result of our Textbyte application is as follow:

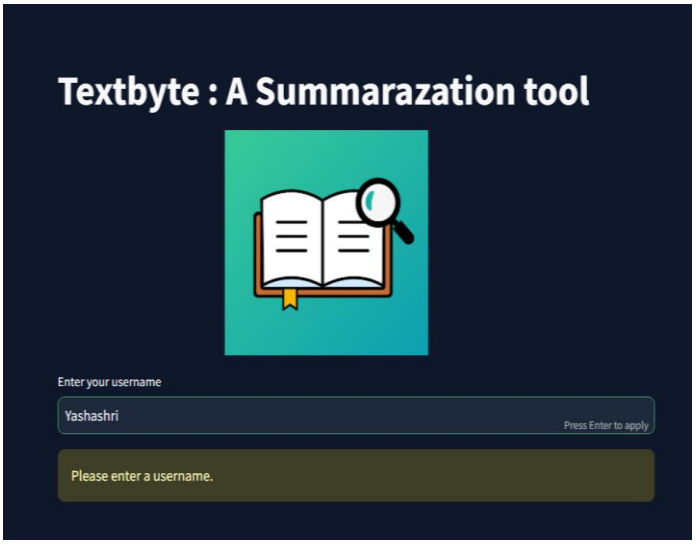

Fig.4.1: UI of Application

The user interface (UI) for our project starts with an initial field prompting the user to enter their username. After entering the username, another field, a drop-down menu, becomes enabled. This drop-down menu allows users to select the category of news they are interested in, such as trending news, favorite topics, search topic and view bookmark.

Once a category is selected, the UI will display relevant news items. Each news item will have options such as "Bookmark" to save it for later reference. When a news item is bookmarked, the UI will display the date and time it was published. Additionally, users can click on "Read More" to view the full news article.

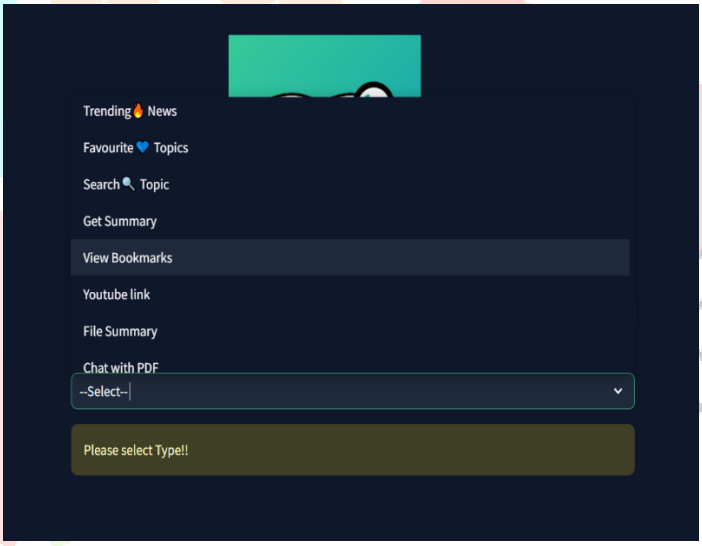

Fig.4.2: Dropdown menu to select category

Once user select get summary you have to provide some article to get the summary of it in extractive and abstractive format. For YouTube video summarization user needs to provide video link which is in English language only. It get the summary of that video. Chat with pdf option is integrated with PDF chatbot which state that user can ask a questions to the chatbot related to the pdf.

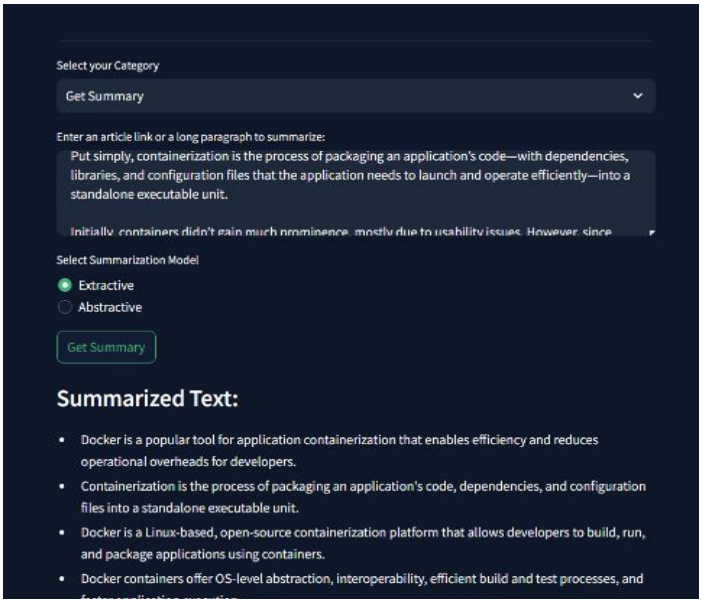

Fig.4.3: Get Extractive Summary

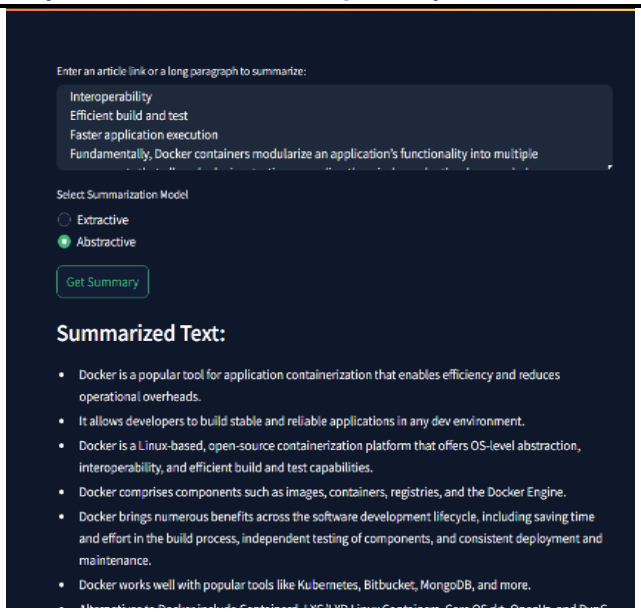

Fig.4.4: Get Abstractive Summary

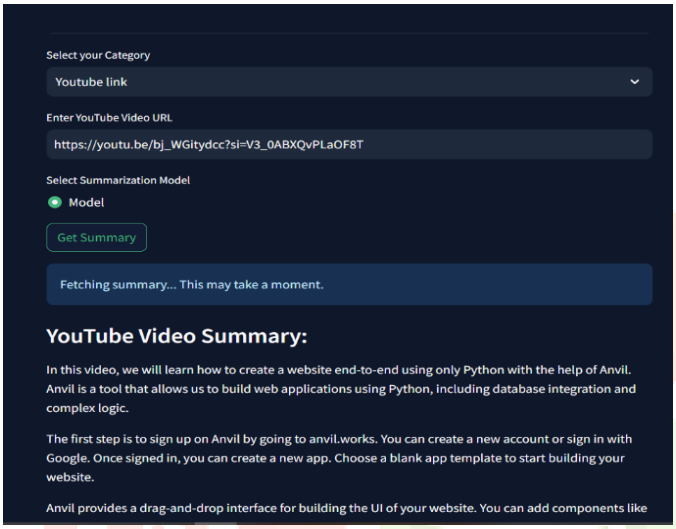

Fig.4.5: YouTube Link

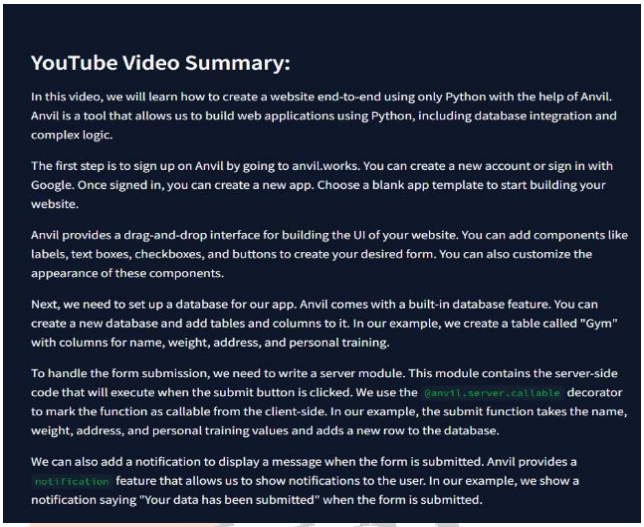

Fig.4.6: Video Summary

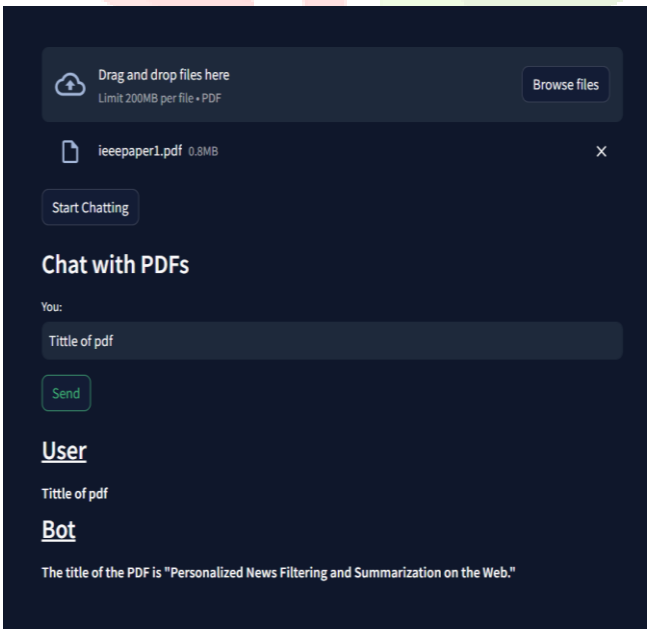

Fig.4.7: Import PDF

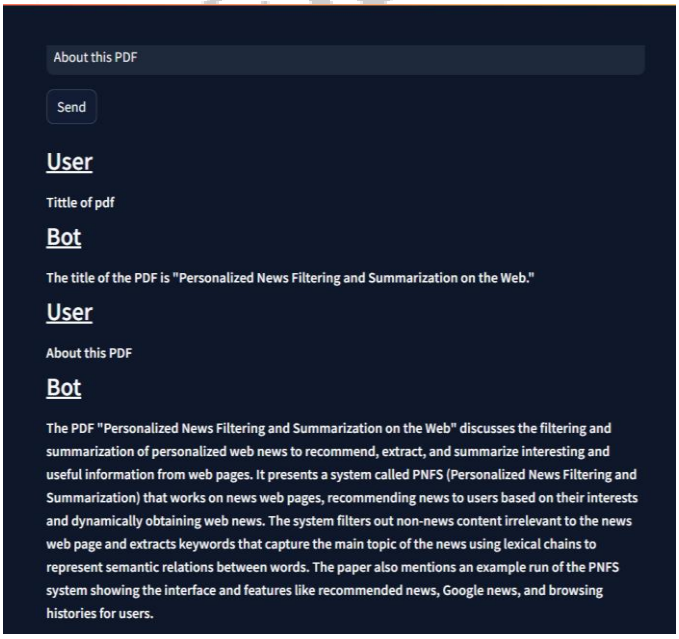

Fig.4.8: Chat with PDF

V. CONCLUSION

Textbyte is an innovative summarization tool that caters to a wide range of content types, including news articles, blog posts, PDF documents, and YouTube videos. This tool effectively summarizes extensive textual content, making it easier for users to understand and retain information.

The summarization feature of Textbyte supports various content types, providing concise summaries that maintain the context and essential points of the original content. The tool is particularly useful for users who want to quickly grasp the main ideas of lengthy articles or news pieces.

Textbyte's YouTube video summarization feature is a unique offering, allowing users to input a YouTube video link, and the tool generates a summary of the video's content. This feature is particularly useful for users who want to quickly understand the content of a video without watching the entire video.

Another notable feature of Textbyte is its PDF chatbot, which allows users to upload a PDF document and ask questions related to its content. The chatbot then searches the PDF for the relevant information and provides a concise answer, making it an effective tool for quick information retrieval from PDF documents.

VI. FUTURE SCOPE

Multilingual Support: Expanding Textbyte's capabilities to summarize text in multiple languages would broaden its user base and make it more useful in global contexts.

Domain-specific Summarization: Tailoring Textbyte for specific domains such as legal documents, medical literature, or technical reports can provide more specialized and accurate summaries for professionals in those fields.

Integration with Knowledge Graphs: Leveraging knowledge graphs and semantic networks to enrich summarizations with related concepts, entities, and relationships can provide deeper insights and context to users.

Real-time Summarization: Implementing real-time summarization capabilities for live events, or social media streams can keep users informed with concise, up-to-date information.

Interactive Summarization: Allowing users to interact with the summarization process by providing feedback or refining summaries based on their preferences can improve the tool's effectiveness over time.

Privacy and Security: Ensuring robust privacy and security measures to protect sensitive information, especially when summarizing confidential documents or personal data, will be crucial for user trust and adoption.

REFERENCES

[1] Sulochana Devi, Rahul Nadar, Tejas Nichat, Alfredprem Lucas, "Abstractive Summerizer for YouTube Videos," published on S. Tamane et al. (Eds.): ICAMIDA 2022, ACSR 105, pp. 431–438, 2023.

[2] Karthik Srikanth "DevOps for Cloud Computing: An Overview," International Journal of Engineering Applied Sciences and Technology, 2022 Vol. 6, Issue 10, ISSN No. 2455-2143, Pages 195-201 Published Online February 2022 in IJEAST (http://www.ijeast.com).

[3] M. F. Mridha, Aklima Akter Lima, Kamruddin Nur, Sujoy Chandra Das, Mahmud Hasan, Muhammas Mohsin Kabir, "A Survey of Automatic Text Summarization: Progress, Process and Challenges" Published on IEEE Access on November 22, 2021.

[4] K. G. Kharade, S. V. Katkar, N. S. Patil, V. R. Sonawane, S. K. Kharade, T. S. Pawar, R. K. Kamat, "Text Summarization of an Article Extracted from Wikipedia Using NLTK Library," Published on Springer Nature Switzerland AG 2021 M. Singh et al. (Eds.): ICACDS 2021, CCIS 1441, pp. 195–207, 2021.

[5] David Reis, Bruno Piedade, Filipe f. Correia João Pedro Dias,Ademar Aguiar, "Developing Docker and Docker-Compose Specifications: A Developers' Survey," published on December 22, 2021

[6] Minh-Tien Nguyen, Van-Chien Nguyen, Huy-The Vu, Van-Hau Nguyen "Transformer-based Summarization by Exploiting Social Information" Published at International Conference on Knowledge and Systems Engineering 2020 12th

[7] Kai Jiang, Xi Lu, "Natural Language Processing and Its Applications" Published on IEEE standards 2020

[8] Laith Abualigah, Mohammad Qassem Bashabsheh, Hamzeh Alabool and Mohammad Shehab, "Text Summarization: A Brief Review" Published on December 02, 2019.

[9] Prathiba Jha, Rizwan Khan, "A Review Paper on DevOps: Beginning and More to Know" Article in International Journal of Computer Applications · June 2018

[10] Chellamalla Mamatha, S C V S L S Ravi Kiran, "Implementation of DevOps Architecture in the project development and deployment with help of tools," published on International Journal of Scientific Research in Computer Science and Engineering Vol.6, Issue.2, pp.87-95, April (2018)

[11] Xindong Wu, Fei Xie, Gongqing Wu Wei Ding, "Personalized News Filtering and Summarization on the Web," presented at the IEEE International Conference on Tools with Artificial Intelligence 2011 23rd

[12] Xindong Wu, Gong-Qing Wu, Fei Xie, Zhu Zhu, and Xue-Gang Hu, Hao Lu and Huiqian Li,, "News Filtering and Summarization on the Web," presented at the IEEE intelligent systems Published by the IEEE Computer Society 1541-1672/10/ 2010

[13] Saeedeh Gholamrezazadeh, Mohsen AminiSalehi, Bahareh Gholamzadeh "A Comprehensive Survey on Text Summarization Systems" Published on IEEE standards 2009.