

---

## Advanced Computing Training School

Centre for Development of Advanced Computing  
Pashan, Pune - 411008

# NewsTracker

24<sup>th</sup> June 2024

**Members:** Manasi Malge, Ameya Bhawsar, Kunal Kurve, Pranav Gaddi, Shrinivas Jawade

## INTRODUCTION

### 1. Overview:

NewsTraker is an innovative Flask web application designed to revolutionize the way users consume news. By providing concise summaries of newspaper articles and accurate answers to user queries, NewsTraker ensures that users can stay informed efficiently and effectively. This application harnesses the power of web scraping, natural language processing (NLP), and advanced machine learning models to deliver an exceptional user experience.

### 2. Goals:

**Time Efficiency:** NewsTraker automates the process of retrieving, summarizing, and answering questions about news articles, saving users valuable time and allowing them to stay informed with minimal effort.

**Enhanced Comprehension:** By combining advanced NLP techniques with powerful language models, NewsTraker provides not only summaries but also clear and accurate answers to specific questions, thereby enhancing users' understanding of the news.

## ARCHITECTURE AND DESIGN

### 1. Pre-requisites:

Python 3, Flask, Requests, BeautifulSoup4, Transformers (Hugging Face), NLTK, langchain\_community, langchain\_text\_splitters, langchain\_core.

### 2. Installation Guide:

#### 1. Install the required libraries:

```
pip install flask requests beautifulsoup4 transformers
```

```
pip install nltk langchain_community langchain_text_splitters langchain_core
```

#### 2. Ensure index.html is in the templates folder and all files are in the same directory:

```
project-directory/
```

---

```
|— app.py
|— templates/
|   |— index.html
```

### 3. Run the Flask Application:

```
python app.py
```

### 4. Open the generated link in your web browser:

## IMPLEMENTATION DETAILS

### 1. Import Necessary Libraries and Modules:

**Flask:** Used to create the web application.

**Requests:** For making HTTP requests to fetch webpage content.

**BeautifulSoup:** For parsing HTML content during web scraping.

**Transformers:** BERT model and pipeline for question answering.

**LangChain:** Framework for document retrieval, embeddings, and other NLP tasks.

**NLTK:** Natural Language Toolkit for stopwords removal and tokenization.

### 2. Download Necessary NLTK Data:

Downloads stopwords and punkt tokenizer models required for text processing.

### 3. Initialize the Flask Application:

Creates a Flask application instance.

### 4. Define Utility Functions:

**remove\_stopwords(text):** Removes stopwords from the input text to focus on significant words.

**scrape\_search\_results(url):** Fetches and cleans text from a webpage by removing HTML tags and stopwords.

### 5. Define Document Class:

Represents a document with its content and metadata.

### 6. Load BERT Model and Tokenizer:

Initializes the BERT model and tokenizer for question answering.

### 7. Initialize Text Splitter and Vector Database:

Splits long text documents into smaller chunks for better processing.

Placeholder for the vector database.

---

## 8. Initialize Local Language Model:

Initializes a local large language model for generating responses.

## 9. Define Prompt Templates and QA Chain:

Defines a prompt template for generating alternative questions to enhance document retrieval.

Placeholders for the retriever and QA chain.

## 10. Define Routes:

### The '/' route:

Renders the home page of the web application.

### The '/summarize' route:

Accepts a URL from the form submission.

Scrapes and cleans the article text.

Splits the text into smaller chunks.

Creates a vector database for document retrieval.

Initializes the retriever and QA chain.

Generates a summary of the article text.

### The '/ask' route:

Accepts a question from the form submission.

Uses the QA chain to retrieve and generate an answer.

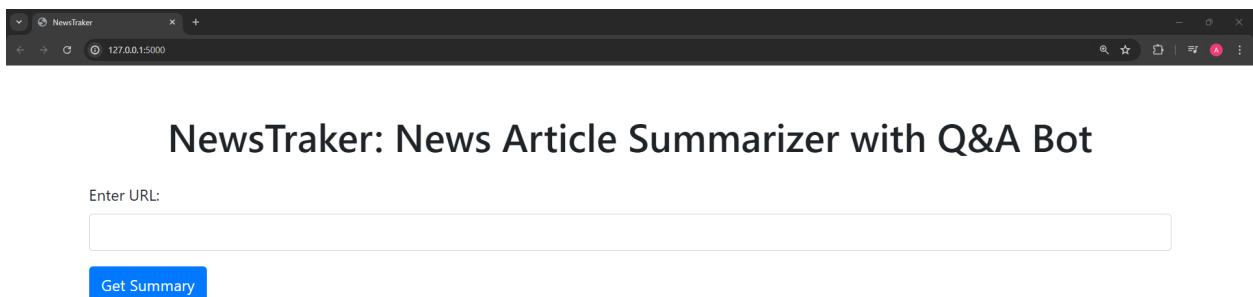
Returns the answer as a JSON response.

## 11. Run the Flask Application:

Runs the Flask web application in debug mode.

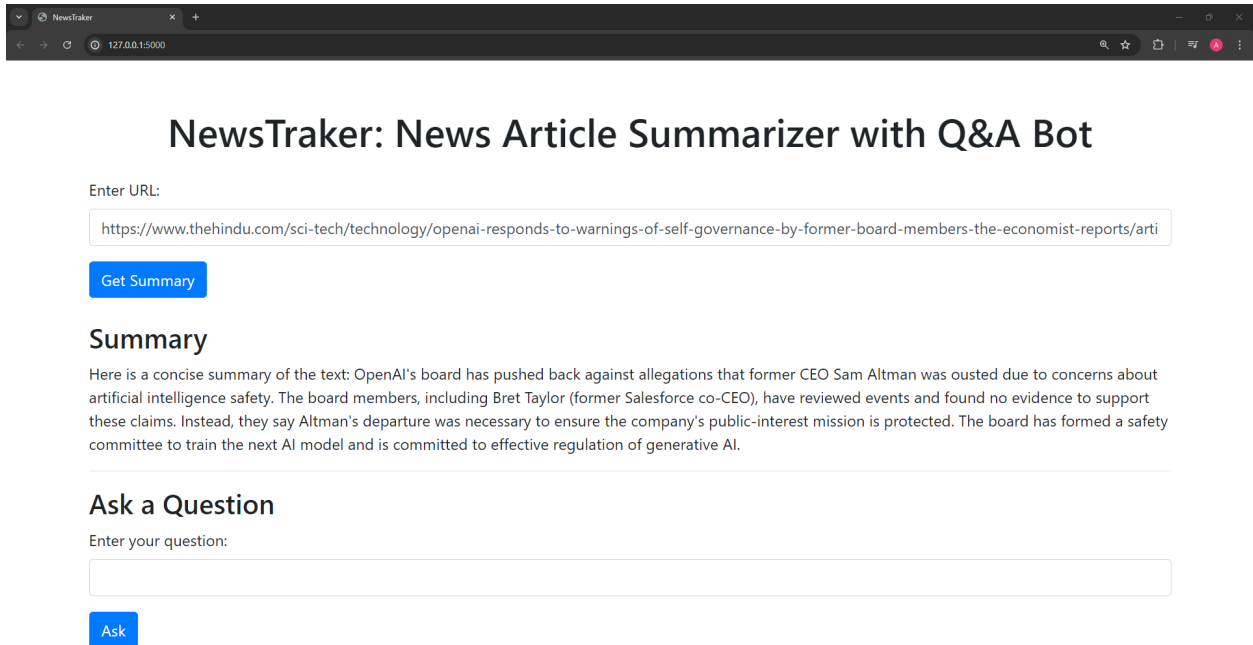
## OUTPUT

### 1. User has to paste the link in the given tab below



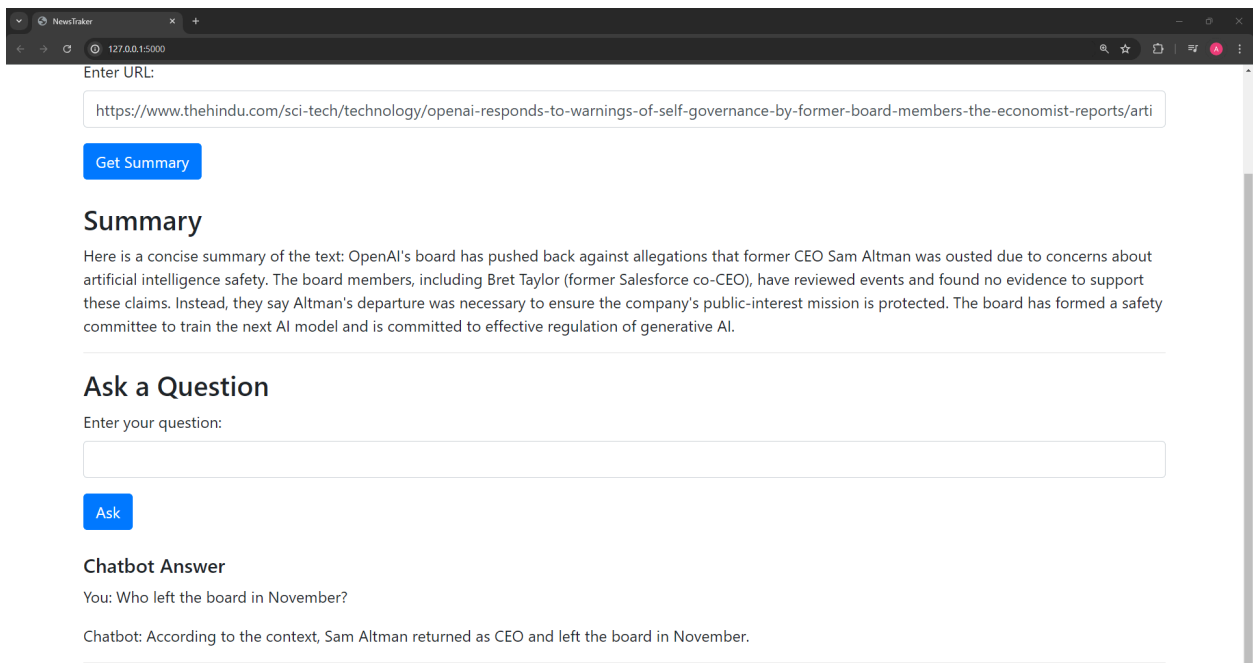
The screenshot shows a web browser window with the title 'NewsTraker'. The address bar displays '127.0.0.1:5000'. The main content area has the heading 'NewsTraker: News Article Summarizer with Q&A Bot'. Below the heading, there is a text input field with the placeholder text 'Enter URL:'. Underneath the input field is a blue button labeled 'Get Summary'.

## 2. After clicking on Get summary, the summary will be generated



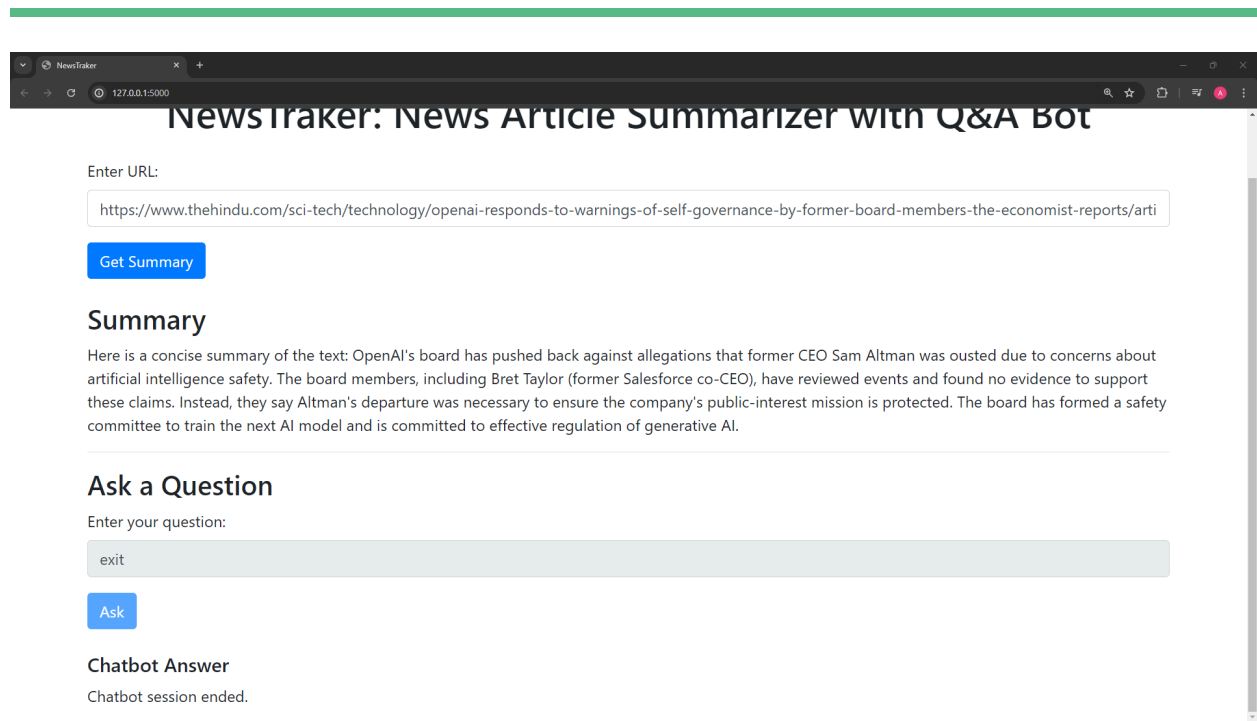
The screenshot shows a web browser window with the title 'NewsTraker'. The address bar displays '127.0.0.1:5000'. The main heading is 'NewsTraker: News Article Summarizer with Q&A Bot'. Below this, there is a section 'Enter URL:' with a text input field containing the URL 'https://www.thehindu.com/sci-tech/technology/openai-responds-to-warnings-of-self-governance-by-former-board-members-the-economist-reports/arti'. A blue button labeled 'Get Summary' is positioned below the input field. Underneath, the 'Summary' section displays a concise summary of the text: 'OpenAI's board has pushed back against allegations that former CEO Sam Altman was ousted due to concerns about artificial intelligence safety. The board members, including Bret Taylor (former Salesforce co-CEO), have reviewed events and found no evidence to support these claims. Instead, they say Altman's departure was necessary to ensure the company's public-interest mission is protected. The board has formed a safety committee to train the next AI model and is committed to effective regulation of generative AI.' Below the summary is another section 'Ask a Question' with a text input field and a blue button labeled 'Ask'.

## 3. User can ask the questions about the article and can interact with the chat bot



This screenshot shows the same web application as the previous one, but with an additional section at the bottom. The 'Ask a Question' section now includes a text input field and a blue button labeled 'Ask'. Below this, a new section titled 'Chatbot Answer' is visible. It contains two lines of text: 'You: Who left the board in November?' and 'Chatbot: According to the context, Sam Altman returned as CEO and left the board in November.'

## 4. To end the session user can enter 'exit'



## FUTURE SCOPE

**Expanding Source Coverage:** Increasing the range of news sources and ensuring coverage of diverse topics can make the application even more valuable to users.

**Improving Summarization Algorithms:** Continuously refining the summarization algorithms to handle more complex articles and generate even more concise summaries.

**Advanced Personalization:** Implementing user-specific personalization features, such as customized news feeds and preference-based summaries.

**Multilingual Support:** Extending the application to support multiple languages, thereby catering to a global audience.

**Implement Pebblo Classifier:** It is a BERT-based model, fine-tuned from distilbert-base-uncased, targeting RAG (Retrieve-And-Generate) applications. It classifies text into categories, streamlining document classification processes.

## CONCLUSION

NewsTraker exemplifies the significant potential of NLP in addressing real-world challenges. By automating the retrieval, summarization, and answering processes, it enhances the efficiency and quality of news consumption. As NLP technologies continue to evolve, applications like NewsTraker will become increasingly sophisticated, offering even greater value to users worldwide. This project not only highlights the current capabilities of NLP but also sets the stage for future innovations in the field of news and information processing.