Article Summarizer Documentation

Pratysh Ranjan Pani

1 Introduction

The "Article Summarizer" is a Python program that uses various libraries like NLTK, TextBlob, and Newspaper to extract and summarize information from a given web article. This graphical user interface (GUI) application is built using the Tkinter library, making it user-friendly and accessible for individuals who want to quickly obtain key details and sentiment analysis of online articles.

2 Features

The Article Summarizer provides the following features:

1. **Title Display**: Displays the title of the article.

2. **Author Display**: Shows the author(s) of the article.

3. **Publication Date Display**: Presents the publication date of the article.

4. **Summary Display**: Provides a summary of the article's content.

5. **Sentiment Analysis Display**: Analyzes the sentiment of the article's content, indicating whether it's positive, negative, or neutral.

6. **URL Input**: Allows users to input the URL of the article they want to summarize.

7. **Summarize Button**: Initiates the summarization process when clicked.

3 Code Overview

The code is divided into two main parts: the GUI creation using Tkinter and the article summarization process using NLTK, TextBlob, and Newspaper. Below is an overview of the code's structure:

3.1 Importing Libraries

```
1  import nltk
2  from textblob import TextBlob
3  from newspaper import Article
4  import tkinter as tk
```

The code starts by importing the necessary libraries, including NLTK for natural language processing, TextBlob for sentiment analysis, Newspaper for article parsing, and Tkinter for creating the GUI.

3.2 `summarize` Function

```
1 def summarize ():
2     # ...
```

This function is called when the "Summarize" button is clicked. It performs the following steps:

- Retrieves the URL entered by the user.

- Downloads and parses the article using the Newspaper library.

- Extracts information such as title, author, publication date, and summary from the article.

- Performs sentiment analysis on the article's text.

- Displays the extracted information and sentiment analysis results in the respective Text widgets in the GUI.

3.3 GUI Creation

```
1 root = tk.Tk ()
2 root.title ("Article Summarizer")
3 root.geometry ("1200x600")
```

This section creates the main GUI window using Tkinter. It sets the window's title and dimensions.

3.4 Text Widgets

```
1 title = tk.Text (root, height =1, width =140)
2 # ...
```

Several Text widgets are created to display the article's title, author, publication date, summary, and sentiment analysis results. These widgets are initially set to "disabled" mode to prevent user input.

3.5 Input URL

```
1 utext = tk.Text (root, height =1, width =140)
2 utext.pack ()
```

A Text widget is provided where the user can input the URL of the article they want to summarize.

3.6 "Summarize" Button

```
1 btn = tk.Button (root, text ="Summarize", command =summarize )
2 btn.pack ()
```

A button labeled "Summarize" is added to the GUI. When clicked, it triggers the `summarize` function to process the entered URL.

3.7 Main Loop

```
1 root.mainloop ()
```

The Tkinter main loop is started, which keeps the GUI running and responsive to user interactions.

4   Usage

To use the "Article Summarizer," follow these steps:

1. Run the script, and the GUI window will appear.

2. Enter the URL of the article you want to summarize in the provided "URL" Text widget.

3. Click the "Summarize" button.

4. The article's title, author, publication date, summary, and sentiment analysis will be displayed in their respective Text widgets.

5   Dependencies

The "Article Summarizer" relies on the following Python libraries and frameworks:

- **nltk**: Used for natural language processing tasks.

- **textblob**: Employed for sentiment analysis of textual data.

- **newspaper**: Enables web scraping and article parsing from online sources.

- **tkinter**: Provides the graphical user interface for the application.

6   Note

- Ensure an internet connection is available as the tool fetches articles from online sources.

- Be aware that web scraping and user agent strings used in the application may be subject to website policies and restrictions.

7   Conclusion

The "Article Summarizer" offers a user-friendly interface to quickly extract and summarize information from web articles. With features such as title display, author information, publication date, summary generation, and sentiment analysis, it provides users with valuable insights into online content. The tool's simple and intuitive interface makes it accessible to a wide range of users for efficient article summarization.