Personalized News Aggregator

Project Overview

This project aims to build a Personalized News Aggregator that collects articles from multiple news sources, categorizes them into different topics using NLP techniques, and serves the collected data via a REST API. The API provides endpoints to retrieve, filter, and search articles. The project is divided into three main parts: News Scraper, Content Categorization, and REST API Development.

Tech Stack

- **Programming Language:** Python
- **Web Scraping:** Selenium, Requests, Webdriver_manager, BeautifulSoup
- **NLP:** spaCy for content categorization
- **Backend:** FastAPI, uvicorn
- **API Testing:** Postman
- **Data Storage:** CSV (news_articles.csv)
- **Additional Modules:** os

Features

Part 1: News Scraper

- **Objective:** Scrape news articles from multiple sources and collect data including:
    - Title
    - Summary (first few sentences)
    - Publication Date
    - Source Name
    - URL
- **Tools:**
    - **Selenium:** For navigating websites and handling dynamic content.
    - **Requests:** To fetch static web pages.
    - **BeautifulSoup:** For parsing HTML and extracting the required information.

    - **Webdriver_manager:** To automatically manage browser drivers.
- **Output:** The scraped data is stored in a CSV file named news_articles.csv with the following columns:
    - Title, Summary, Publication Date, Source, and URL.

Part 2: Content Categorization

- **Objective:** Categorize articles into topics like politics, technology, sports, etc., using Natural Language Processing (NLP).
- **Tools:**
    - **spaCy:** A powerful NLP library used for tokenization, part-of-speech tagging, and categorization.
    - The articles are processed to determine their respective categories based on the content.
- **Output:** The CSV file news_articles.csv is updated to include an additional Category column.

Part 3: REST API Development

- **Objective:** Develop a REST API to serve the scraped and categorized news articles.
- **Tools:**
    - **FastAPI:** A modern, fast web framework for building APIs.
    - **uvicorn:** For running the FastAPI server.
    - **Postman:** For API testing and development.
- **API Endpoints:**
    - **GET /articles:** Retrieve all articles with optional filters (date range, category).
    - **GET /articles/{id}:** Retrieve a specific article by its ID.
    - **GET /search:** Search for articles by keywords.
- **Output:** The API returns data in JSON format with proper filtering and search functionality.

Installation and Setup

1. **Clone the repository:**

```
git clone https://github.com/yourusername/Personalized-News-Aggregator
```

```
.git
```

2. **Navigate to the project directory:**

```
cd Personalized-News-Aggregator
```

3. **Install the required dependencies:**

```
pip install -r requirements.txt
```

4. **Run the news scraper to collect articles:**

```
python bbcNews_Scraper.py

python thetimesofindiaNews_Scraper.py
```

5. **Content Categorizaion using NLP**

```
python contentCategorization.py
```

6. **Start the FastAPI server:**

```
uvicorn main:app --reload
```

API Usage

1. **Retrieve all articles**
    - **Endpoint:** GET /articles
    - **Description:** Retrieves all the news articles stored in the database with optional filters.
    - **Filters:** You can filter articles by:
    - **category:** Filter by news category.
    - **date_range:** Specify a start and end date for filtering. Example:

```
curl -X GET "http://127.0.0.1:8000/articles"
```

2. **Retrieve an article by ID**
    - **Endpoint:** GET /articles/{id}
    - **Description:** Retrieves a specific article based on the provided ID. Example:

```
curl -X GET "http://127.0.0.1:8000/articles/1"
```

3. **Search for articles**

- **Endpoint:** GET /search
- **Description:** Searches articles by keywords found in the title or summary. Example:

```
curl -X GET "http://127.0.0.1:8000/search?keyword=climate"
```

Screenshots and Videos

- Checkout video of the working API endpoints is included in the below drive link:

  https://drive.google.com/file/d/1oaWsy1wsqeoPvw7zhojcVsivqGr4b33F/view?usp=sharing

Postman Collection

- A Postman collection for testing the API is included in the repository as postman_collection.json.
- To use:
  - Import the collection into Postman.
  - Use the available requests to test the API endpoints.

Author

- Koyalkar GopiKrishna.
- **Email:** koyalkargopikrishna@gmail.com