Google News Scraper Pipeline

Overview

This project is an automated web scraping pipeline that extracts top news articles from Google News, including headlines, article URLs, and corresponding images. The extracted data is stored in a MongoDB database for further analysis. The pipeline is modular and consists of multiple Python scripts that handle different stages of the process: scraping the homepage, extracting "Top Stories," handling lazy-loaded content, storing data in a structured database, and ensuring de-duplication. The pipeline is designed for automation and can be scheduled using cron jobs for continuous data collection.

Project Structure

google-news-scraper

- config.ini # Configuration file for scraper settings
- mainf.py # Orchestrates scraping and database storage
- scraper.py # Handles web scraping using Selenium
- mongodb.py # Stores extracted data into MongoDB
- pipeline.log # Logs execution details
- README.md # Project documentation

Installation & Setup

1. Clone the Repository

```
git clone https://github.com/your-repo/google-news-scraper.git
cd google-news-scraper
```

2. Set Up MongoDB

Ensure MongoDB is installed and running on localhost:27017.

```
sudo systemctl start mongod  # For Linux
```

3. Configure **config.ini**

Modify config.ini to adjust scraping parameters:

```
[WEBSCRAPING]
google_news_url = https://news.google.com
top_stories_id = i11
element_class_name_1 = .IBr9hb
element_class_name_2 = .IFHyqb
thumbnail_class_name = .Quavad.vwBmvb
article_class_name_1 = .gPFEn
```

article_class_name_2 = .JtKRv
article_class_url = .WwrzSb
article_class_time = .hvbAAd

Modules

1. main.py (Pipeline Orchestration)

- Loads configuration from config.ini.
- Initializes the Selenium WebDriver.
- Calls fetch_news_details() from scraper.py to scrape articles.
- Saves the scraped data to news_data.csv.
- Calls store_data_in_mongodb() from mongodb.py to insert data into MongoDB.

2. scraper.py (Web Scraping)

- Uses Selenium to extract headlines, article URLs, timestamps, and images.
- fetch_news_details(): Scrapes article metadata and images.
- retrieve_top_stories_url(): Extracts the URL of the 'Top Stories' section.

3. mongodb.py (Database Handling)

- Connects to MongoDB.
- Stores metadata in gnews_metadata and image data in gnews_images.
- Ensures unique headlines to prevent duplicates.

How to Run

1. Install dependencies:

   ```
   pip install selenium pymongo pandas requests configparser
   ```
2. Start MongoDB server.
3. Run the script:

   ```
   python main.py
   ```

Troubleshooting

- If MongoDB isn't running, start it using:

  ```
  sudo systemctl start mongod
  ```
- If Chrome WebDriver fails, update it:

  ```
  pip install --upgrade chromedriver-autoinstaller
  ```

Contributing

Feel free to submit issues or pull requests to enhance the scraper.

Future Enhancements

- Implement advanced de-duplication based on article similarity rather than just headlines.
- Extend database support to PostgreSQL or MariaDB for relational storage.
- Integrate a scheduling system like Apache Airflow for better workflow management.

License