Assignment 5.12

Individual Reflections (Sprint 1)

Team Headline Grabber

Sheena Munjal (smunjal)

What you planned to do?

HG5: Unit test for new command line option to include/exclude certain news sources from application default settings

HG18: Unit test for implementation of parsing HTML data from WAP (Washington Post)

HG21: Implement the progress bars for each step of the news pipeline

What you did not do?

None

What problems you encountered?

1. Setting up the project on the IDE

Files you worked on?

1. **For user story HG5:**
   tests/test_filter_sites.py

2. **For user story HG18:**

tests/test_scrape_text_wap.py

3. **For user story HG21:**

src/headline_grabber/pipeline_steps/classify_subject.py

src/headline_grabber/pipeline_steps/display_report.py

src/headline_grabber/pipeline_steps/group_by_similarity.py

src/headline_grabber/pipeline_steps/pipeline_step.py

src/headline_grabber/pipeline_steps/prepare_for_display.py

src/headline_grabber/pipeline_steps/score_sentiment.py

src/headline_grabber/pipeline_steps/scrape_text.py

src/headline_grabber/pipeline_steps/text_similarity.py

What you accomplished?

Objectives and Achievements

1. **HG5- Unit test for new command line option to include/exclude certain news sources from application default settings: -**

- The test ensures that the application correctly handles user input for the include and exclude options. It verifies that the user can specify valid news sources and that the application processes these inputs accurately.
- The test validates that the application correctly filters the news sources based on the specified include or exclude options. It ensures that only the desired sources are considered when generating the headline report, providing users with relevant content.
- The test encourages better code quality by ensuring that the logic for handling the command-line options is robust and well-implemented.

2. **HG18- Unit test for implementation of parsing HTML data from WAP (Washington Post): -**

- This unit test is essential for validating the implementation of the HTML parsing functionality specific to The Washington Post (WAP) within our headline collection and summarization project.
- The test ensures that the _parse_headline method in the ScrapeText class accurately extracts and cleans relevant data from the HTML structure provided by WAP.
- It verifies that the parsed headline's link, title, and description match the expected values derived from the sample HTML (STUB_HTML).
- By simulating the parsing process with predefined selectors and a dummy HTML snippet, the test confirms the robustness and reliability of the parsing logic

3. **HG21- Implement the progress bars for each step of the news pipeline: -**

- Long-running processes, such as data processing, web scraping, or model inference, can take significant time to complete. Progress bars give users visibility into the current state

of the operation, making them aware of how much work has been done and how much is left.

- Without progress bars, users might be left wondering if the program is still running or if it has stalled or crashed.
- Implemented "tqdm" in this user story which is a Python library that enables the creation of progress bars for loops and iterative processes. When wrapped around an iterable, tqdm displays a dynamic progress bar in the console that updates in real-time, showing the current progress