Report

PubMed Paper Fetcher

1. Introduction

The **PubMed Research Fetcher** is a Python-based tool designed to fetch research papers from PubMed based on user-defined queries, specifically targeting papers related to pharmaceutical and biotech industries. The tool provides functionalities to filter authors affiliated with pharmaceutical companies, save the results to a CSV file, and enable debug mode for detailed logging.

This report outlines the approach taken to build the tool, the methodology used in designing the program, and the results observed during its execution.

2. Approach

The main objective of the **PubMed Research Fetcher** was to provide an efficient way for researchers to access relevant academic papers from PubMed. The approach involved:

1. **Using the PubMed API**: The tool interacts with the PubMed E-utilities API to search for relevant papers based on user queries.
2. **Filtering Authors**: The program filters authors based on company affiliations (pharmaceutical and biotech companies) to identify non-academic contributors.
3. **Saving Results**: The program allows saving the fetched results to a CSV file for further analysis or record-keeping.
4. **Error Handling and Debugging**: Implemented error handling for issues like invalid queries, network problems, and missing data, along with a debug mode to print detailed logs.

3. Methodology

3.1. Code Structure

The code was organized into two main parts:

- pubmed_utils.py: A module that handles the fetching of PubMed IDs and paper details, as well as parsing the API response and identifying authors with non-academic affiliations.
- main.py: The command-line interface (CLI) script that allows users to run the program, enter queries, and save or display the results.

3.2. API Interaction

The **PubMed Research Fetcher** utilizes the **PubMed E-utilities API** to interact with the PubMed database. The following steps are involved in fetching data from the API:

1. **Querying PubMed**: The user enters a search query (e.g., "pharmaceutical research") which is sent to the PubMed API's esearch endpoint to retrieve a list of PubMed IDs.
2. **Fetching Paper Details**: The PubMed IDs are then used to fetch detailed information about each paper from the efetch endpoint. The data returned includes the title, publication date, author names, and affiliations.
3. **Parsing XML Responses**: The responses from the PubMed API are in XML format, which is parsed to extract the necessary information (e.g., author names, affiliations, emails).

3.3. Filtering Non-Academic Authors

The tool identifies non-academic authors based on specific company-related keywords (e.g., "Pharma", "Biotech", "Inc"). These authors are typically affiliated with pharmaceutical and biotech companies. This filtering process helps researchers identify relevant authors for industry-specific papers.

3.4. Saving Results to CSV

The paper details, including titles, publication dates, author affiliations, and emails, are saved to a CSV file for further analysis. The user can specify the filename through a command-line argument.

3.5. Debug Mode and Error Handling

The program includes robust error handling:

- **Invalid queries** (e.g., empty or whitespace queries) are caught and a message is displayed.
- **Network issues** or API failures are handled by retrying the request up to 3 times.
- The program prints debug logs when the -d flag is used, providing additional insights into the program’s execution.

4. Results

4.1. Performance

The program performs efficiently for typical queries, retrieving paper details and saving them to a CSV file within seconds. Multiple queries can be processed in sequence without significant delays.

4.2. Accuracy

The accuracy of the search results is dependent on the PubMed API’s response. In tests, the tool successfully retrieved relevant research papers for queries like "pharmaceutical research" and "biotech companies". The filtering of non-academic authors based on company affiliations also worked as expected.

4.3. Debug Mode

When running the program in debug mode (-d), detailed logs were generated, showing the API responses and steps taken by the program to process each request. This helped confirm that the program is functioning as intended and provided valuable insights for debugging any issues.

4.4. Handling of Edge Cases

- Invalid or empty queries were caught and the user was prompted with a clear error message.
- If the API failed to return results or if the response format was incorrect, the program raised a ValueError and handled retries gracefully.

5. Conclusion

The **PubMed Research Fetcher** successfully meets the project's requirements by:

- Fetching research papers from PubMed based on user queries.
- Filtering authors affiliated with pharmaceutical and biotech companies.
- Saving the results to a CSV file.
- Providing error handling and debug mode for smooth user experience.

The tool proves to be an effective resource for researchers in the pharmaceutical and biotech industries, helping them quickly gather relevant research papers and authors. Future improvements could include adding more filters (e.g., by year) and expanding to other research databases.