Research Paper Fetcher and Filter

Prepared by: Nikhil Shashikant Shinde
Email: nikhilshinde2598@gmail.com
Mobile: 9834672510

1. Introduction

The task was to develop a Python-based program that fetches research papers from the PubMed API, filters papers based on their affiliation with pharmaceutical or biotech companies, and returns the filtered results as a CSV file. The program must support PubMed's full query syntax and should display usage instructions with various options.

2. Approach and Methodology

2.1 Problem Breakdown

The problem was broken into several parts:
1. Fetch Research Papers: The program will fetch research papers using the PubMed API by supporting flexible querying.
2. Filter Based on Affiliation: The program must identify and filter research papers based on author affiliation with pharmaceutical or biotech companies.
3. Output the Results: The filtered results will be written to a CSV file with specific columns like PubmedID, Title, Publication Date, Non-academic Author(s), Company Affiliation(s), and Corresponding Author Email.
4. Command-Line Interface (CLI): The program will accept a query as a command-line

2.2 Design and Implementation

Fetching Data from PubMed API:
The program uses HTTP requests to fetch data from the PubMed API.
The endpoint https://api.ncbi.nlm.nih.gov/lit/ctxp/v1/pubmed/?format=json&query=
was used to query the PubMed database. The query was URL-encoded to handle special characters and spaces.

Filtering Non-academic Authors and Company Affiliations:
To identify non-academic authors, the program uses basic heuristics, such as looking for keywords in author names or email addresses. For company affiliation, a check is performed to see if any author’s affiliation contains terms like 'pharmaceutical' or 'biotech'.

CSV Output:
The results are written to a CSV file using the Apache Commons CSV library. The CSV columns include:
- PubmedID
- Title
- Publication Date
- Non-academic Author(s)
- Company Affiliation(s)
- Corresponding Author Email

Command-Line Arguments:
The program supports the following command-line arguments:
-h or --help: Displays usage instructions.
-d or --debug: Prints debug information during execution.
-f or --file: Specifies the output file name. If this option is not provided, the program outputs to the console.

2.3 Error Handling

The program includes error handling for invalid queries, failed API requests, or missing data. For missing data, placeholders are used to avoid null values.

3. Results

3.1 Example Run

Here is an example of how the program would work:

1. The user runs the following command:

```
java com.example.pubmed.Main 'biotech cancer' -f output.csv
```

2. The program fetches the relevant papers from the PubMed database, applies the filter for pharmaceutical and biotech company affiliations, and writes the results to output.csv.

3. The output CSV might look like this:

PubmedID	Title	Publication Date	Non-academic Author(s)	Company Affiliation(s)	Corresponding Author Email
12345678	Biotech Innovations in Cancer	2025-01-12	Nikhil S. Shinde	PharmaCorp, BioTech.	nikhil@gmail.com
23456789	Cancer Research by Biotech Firms	2024-09-05	Prathmesh pawar	MedPharm Labs	prathmesh@gmail.com

3.2 Debug Information (Optional)

If the user passes the -d flag, the following debug information would be printed during execution:

Fetching from URL:
https://api.ncbi.nlm.nih.gov/lit/ctxp/v1/pubmed/?format=json&query=biotech+cancer
API Response: { 'papers': [{ 'pmid': '12345678', 'title': 'Biotech Innovations in Cancer', 'pubdate': '2025-01-12', ... }] }

4. Code Organization

The code is divided into the following modules:
1. Paper Class: Represents a research paper with properties such as PubMed ID, title, publication date, authors, affiliations, and corresponding author email.
2. PaperFetcher Class: Handles fetching papers from the PubMed API and parsing the response.

3. CSVWriter Class: Responsible for writing the fetched data to a CSV file.
4. Main Class: Contains the entry point of the program and handles command-line arguments.

5. Tools and Libraries
- Apache Commons CSV: Used for writing data to a CSV file.
- Jackson (ObjectMapper): Used for parsing JSON responses from the PubMed API.
- Java HTTP Client (HttpClient): Used to make API requests to PubMed.
- Command-line Argument Parsing: The program accepts arguments for flexibility.

6. Conclusion

The program successfully fetches research papers from the PubMed API, filters them based on pharmaceutical and biotech affiliations, and exports the results to a CSV file. It supports flexible querying, error handling, and command-line options for customization. Future improvements could involve more advanced filtering techniques and more detailed parsing of author information.

7. Future Improvements

- Implement more sophisticated logic to parse author affiliations.
- Use machine learning techniques to better identify non-academic authors and companies.

8. References

- [PubMed API Documentation](https://www.ncbi.nlm.nih.gov/books/NBK3827/)
- [Apache Commons CSV Library](https://commons.apache.org/proper/commons-csv/)
- [Jackson JSON Library](https://github.com/FasterXML/jackson)