PubMed Paper Fetcher: Approach, Methodology, and Results

1. Introduction

This report provides an overview of the approach, methodology, and results of the **PubMed Paper Fetcher** project. The project aims to fetch research papers from PubMed based on user-specified queries, identify papers with at least one author affiliated with a pharmaceutical or biotech company, and return the results as a CSV file.

2. Approach

The approach follows a structured workflow:

1. **Fetching PubMed Data**: Utilize the PubMed API to retrieve relevant research papers based on a given query.
2. **Extracting Relevant Information**: Parse XML responses to extract metadata such as Pubmed ID, title, publication date, author details, and affiliations.
3. **Filtering Non-Academic Authors**: Identify authors affiliated with pharmaceutical or biotech companies using keyword-based heuristics.
4. **Saving Results**: Store the extracted data in a structured CSV file for easy access and analysis.
5. **Command-line Interface**: Provide a CLI tool to automate fetching and filtering papers using user-specified queries.

3. Methodology

3.1 Data Fetching

- The fetch_papers.py module queries PubMed using the esearch API to obtain relevant PubMed IDs.
- The efetch API is then used to retrieve full details of each paper in XML format.

3.2 Data Extraction

- The XML response is parsed using the ElementTree library.
- Metadata such as **PubmedID, title, publication date, authors, affiliations, and emails** is extracted.
- A regex-based method extracts email addresses when available.

3.3 Filtering Non-Academic Authors

- A predefined list of company-related keywords (e.g., "pharma", "biotech", "ltd", "inc", "therapeutics", etc.) is used to identify non-academic authors.
- If at least one author is found with a company-affiliated keyword in their institution, the paper is included in the final output.

3.4 Data Output

- The filtered results are saved in a CSV file containing:
    - **PubmedID**
    - **Title**
    - **Publication Date**
    - **Non-academic Author(s)**
    - **Company Affiliation(s)**
    - **Corresponding Author Email**

- The save_to_csv function handles CSV creation and output validation.

3.5 CLI Integration

- The cli.py script provides a command-line interface with:
  - -h / --help: Displays usage instructions.
  - -d / --debug: Enables debug mode for detailed logging.
  - -f / --file: Allows the user to specify the output filename.

- **4. Results**
- **4.1 Sample Query Execution**
- Using the following command

Using the following command:

```
poetry run get-papers-list "biotech India" -f output.csv
```

4.2 Sample Output (CSV Format)

PubmedID	Title	Publication Date	Non-academic Author(s)	Company Affiliation(s)	Corresponding Author Email
12345678	Drug Discovery in India	2023-06-12	Dr. Raj Kumar	Biocon Pvt Ltd	raj.kumar@biocon.com
87654321	*Advances in Pharma*	2022-11-05	Dr. Meera Das	Cipla Ltd	meera.das@cipla.com

5. Version Control & Deployment

- The project is hosted on **GitHub**: PubMed Paper Fetcher Repo
- **Poetry** is used for dependency management and packaging.
- **Installation & Execution:**

1. Clone the repository:

```
git clone https://github.com/Bonupavankumar/pubmed-paper-fetcher.git
```

2. Navigate to the directory:

```
cd pubmed-paper-fetcher
```

3. Install dependencies:

```
poetry install
```

4. Run the CLI tool:

```
poetry run get-papers-list "your query" -f results.csv
```

6. Conclusion

The **PubMed Paper Fetcher** successfully automates the retrieval, filtering, and storage of research papers with non-academic authors. It provides a user-friendly CLI interface and ensures structured output in CSV format. Future improvements could include:

- Advanced NLP-based affiliation classification.
- Parallel processing for faster API queries.
- Integration with a web interface for broader usability.

Developed by: Bonupavankumar **GitHub Repository:**
https://github.com/Bonupavankumar/pubmed-paper-fetcher