Research Paper Fetcher from PubMed Based on Author Affiliation

Project Overview

This project implements a command-line Python application designed to fetch research papers from PubMed using user-specified queries. The primary goal is to identify and extract papers that include at least one author affiliated with a pharmaceutical or biotech company and output the results in CSV format or to the console. The system is built with modular design principles, robust error handling, and flexibility for extended features.

Objectives

- Enable flexible PubMed queries supporting full query syntax.

- Filter papers based on author affiliations, specifically targeting pharmaceutical and biotech companies.

- Extract key metadata: PubMed ID, Title, Publication Date, Authors, Affiliations, and Corresponding Author Email.

- Provide command-line options for user convenience.

- Maintain code readability, performance, and robustness.

System Architecture

Module (`pubmed\_fetcher.py`)

- Handles all interactions with PubMed API.

- Parses and filters retrieved data.

- Implements heuristics for identifying non-academic authors.

- Provides core functions for data fetching and processing.

Command-line Interface (`main.py`)

- Parses user inputs and options.

- Calls core module functions.

- Manages output formatting and storage.

Project Structure

get-papers-list/

│

├── pubmed\_fetcher.py # Core logic for API interaction and data processing

├── main.py # CLI interface

├── tests/ # Unit tests (optional)

├── README.md # Documentation

├── pyproject.toml # Poetry configuration

Implementation Details

Data Retrieval

- Utilizes PubMed's E-utilities API (Entrez Programming Utilities).

- Supports complex query syntax.

- Retrieves XML data, which is parsed using Python's `xml.etree.ElementTree`.

Data Filtering

- Applies heuristics to identify author affiliations:

- Presence of keywords such as "pharma", "biotech", "laboratories", "company", etc.

- Email domain heuristics (e.g., corporate domains).

- Ensures at least one author is affiliated with a pharmaceutical or biotech entity.

Data Extraction

- PubMed ID (`PMID`)

- Paper Title

- Publication Date

- Author Names (excluding academic affiliations)

- Company Affiliations

- Corresponding Author Email

Output

- CSV file with defined columns.

- Alternatively, print to console if no filename specified.

Command-line Options

| Option | Description | Default |

|---------|--------------|---------|

| `-h` / `--help` | Show usage instructions | N/A |

| `-d` / `--debug` | Enable debug logging | False |

| `-f` / `--file` | Output filename | None (print to console) |

| `--query` / `-q` | PubMed search query | Required |

Technologies & Tools

- \*\*Python 3.8+\*\*: Core language

- \*\*Requests\*\* (`https://requests.readthedocs.io/`): HTTP requests handling

- \*\*Pandas\*\* (`https://pandas.pydata.org/`): Data manipulation and CSV output

- \*\*xml.etree.ElementTree\*\*: XML parsing (built-in)

- \*\*Poetry\*\* (`https://python-poetry.org/`): Dependency management and packaging

- \*\*Git\*\* (`https://git-scm.com/`): Version control

- \*\*GitHub\*\* (`https://github.com/`): Repository hosting

Setup & Usage Instructions

Environment Setup

Clone the repository

git clone https://github.com/yourusername/get-papers-list.git

cd get-papers-list

Install dependencies

poetry install

Running the Program

Example: Search for papers with "cancer" and output to 'results.csv' with debug info

poetry run python main.py --query "cancer" --file results.csv --debug

Command-line Options

- `--query` (`-q`): PubMed search string (mandatory)

- `--file` (`-f`): Output filename (defaults to console if omitted)

- `--debug` (`-d`): Enable debug logs

- `--help` (`-h`): Show help message

Future Improvements

- Enhance heuristics for identifying non-academic authors.

- Support for additional filtering options.

- Parallelize API calls for performance.

- Publish module on TestPyPI for reuse.

Licensing and Contribution

This project is licensed under the MIT License. Contributions are welcome via GitHub pull requests. Please ensure your code adheres to PEP8 standards and includes proper documentation.

References

- [PubMed E-utilities API](https://www.nlm.nih.gov/research/pubmed electronic\_help.html)

- [Poetry Documentation](https://python-poetry.org/docs/)

- [Requests Library](https://requests.readthedocs.io/en/master/)

- [Pandas Documentation](https://pandas.pydata.org/pandas-docs/stable/)