Research Paper Fetcher: Project Report

1. Introduction

This report summarizes the development of a Python-based tool to fetch research papers from PubMed, filter those with pharmaceutical/biotech affiliations, and export results in CSV format. The solution adheres to the specified requirements, including command-line interface (CLI) support, PubMed API integration, and proper output formatting.

---

2. Approach & Methodology

2.1 Problem Breakdown

The task was divided into three main components:

1. **PubMed API Integration**: Fetch papers using PubMed's E-utilities API.
2. **Data Processing**: Filter papers with industry affiliations and extract relevant metadata.
3. **Output Generation**: Export results in CSV format with specified columns.

2.2 Key Design Decisions

- **Modular Architecture**: Separated logic into `api_client`, `processor`, and `models` for maintainability.
- **Robust Filtering**: Used keyword-based heuristics to identify industry affiliations (e.g., "pharma," "biotech").
- **Error Handling**: Implemented graceful failure for API errors, missing data, and invalid queries.
- **Type Safety**: Used Python type hints (`List[str]`, `Optional[Author]`) for better code reliability.
- **Rate Limiting**: Added a delay (0.34s) between API calls to comply with PubMed's guidelines (max 3 requests per second).

2.3 Implementation Steps

1. **PubMed API Client (`api_client.py`)**
   - Used `requests` to interact with PubMed's E-utilities (`esearch`, `efetch`).
   - Parsed XML responses to extract paper metadata (title, authors, affiliations, etc.).
2. **Data Processing (`processor.py`)**
   - Filtered papers based on industry-related keywords in affiliations.

   - Extracted non-academic authors and corresponding author emails.
3. **CLI Integration (`get_papers_list.py`)**
   - Used `argparse` to support command-line arguments (`--query`, `--file`, `--debug`).
   - Enabled CSV output to a file or stdout.
4. **Testing & Validation**
   - Tested with real PubMed queries (e.g., `"cancer AND treatment"`).
   - Verified CSV output correctness (columns: `PubmedID`, `Title`, `Non-academic Authors`, etc.).

---

3. Results

3.1 Functional Output

The program successfully:
✔ Fetches papers from PubMed based on user queries.
✔ Identifies pharmaceutical/biotech affiliations using keyword matching.
✔ Generates a CSV with the required columns:

- `PubmedID`
- `Title`
- `Publication Date`
- `Non-academic Author(s)`
- `Company Affiliation(s)`
- `Corresponding Author Email`

Example CSV Output:

csv

```
PubmedID,Title,Publication Date,Non-academic Author(s),Company Affiliation(s),Corresponding Author Email
12345678,"Novel Cancer Drug Trial",2023-05-15,John Doe; Jane Smith,Genentech;john.doe@genentech.com
```

3.2 Performance

- **API Calls**: Efficient batch fetching of papers (100+ in ~34 seconds due to rate limiting).
- **Filtering**: Near-instant processing after data retrieval.

3.3 Limitations & Future Improvements

- **Affiliation Detection**: Current keyword-based approach may miss some companies. A machine learning (ML) classifier could improve accuracy.

- **Email Extraction**: Relies on parsing affiliations; a more robust method (e.g., regex) could help.
- **Parallel Fetching**: Could improve speed (but must respect PubMed’s rate limits).

---

4. Conclusion

The **Research Paper Fetcher** successfully meets all specified requirements, providing a reliable way to:

1. Query PubMed for research papers.
2. Filter those with industry affiliations.
3. Export structured data for further analysis.

The modular design ensures maintainability, and the CLI interface makes it easy to integrate into automated workflows. Future enhancements could include:

- More sophisticated affiliation detection.
- Support for additional academic databases (e.g., IEEE Xplore, arXiv).

GitHub Repository: https://github.com/keerthireddy36/backendresearchpapers/tree/main