Project Report: PubMed Paper Fetcher

1. Introduction

The PubMed Paper Fetcher is a Python-based tool designed to automate the process of fetching research papers from PubMed based on a user-defined query. The tool extracts key details such as paper titles, publication dates, authors, affiliations, and corresponding author emails. It also identifies non-academic authors and their company affiliations. The results can be saved to a CSV file for further analysis.

This report summarizes the approach, methodology, and results of the project.

---

2. Approach

The project was developed with the following goals in mind:

- **Automation**: Simplify the process of fetching and analyzing research papers from PubMed.
- **Customization**: Allow users to define search queries and filter results based on specific criteria.
- **Data Export**: Enable users to save results in a structured format (CSV) for further analysis.

The tool was built using Python and leverages the PubMed API (`eutils`) for fetching data. Key libraries used include:

- `requests`: For making HTTP requests to the PubMed API.
- `xml.etree.ElementTree`: For parsing XML responses from PubMed.
- `csv`: For saving results to a CSV file.
- `argparse`: For handling command-line arguments.

---

3. Methodology

3.1 Data Fetching

1. **PubMed ID Retrieval**:
    - The tool sends a query to the PubMed `esearch` API to fetch a list of PubMed IDs matching the search criteria.
    - The query is customizable, allowing users to specify keywords, filters, and result limits.
2. **Paper Details Extraction**:

- For each PubMed ID, the tool fetches detailed information using the PubMed `efetch` API.
- The response is in XML format, which is parsed to extract relevant fields such as title, publication date, authors, affiliations, and emails.

3.2 Data Processing

1. **Author and Affiliation Analysis**:
   - The tool identifies non-academic authors by checking affiliations for keywords such as "Inc", "Ltd", "Pharma", etc.
   - Corresponding author emails are extracted using regex patterns from affiliation text.
2. **Publication Date Parsing**:
   - The publication date is parsed from the XML response. If the full date (year, month, day) is unavailable, only the year is extracted.

3.3 Data Export

- The extracted data is saved to a CSV file with the following columns:
  - `PubmedID`
  - `Title`
  - `Publication Date`
  - `Non-academic Author(s)`
  - `Company Affiliation(s)`
  - `Corresponding Author Email`

---

4. Results

4.1 Functionality

The tool successfully:

- Fetches PubMed IDs for a given query.
- Extracts detailed information for each paper.
- Identifies non-academic authors and their affiliations.
- Saves results to a CSV file.

4.2 Example Output

For the query `"cancer research"`, the tool generated the following sample output:

PubmedID	Title	Publication Date	Non-academic Author(s)	Company Affiliation(s)	Corresponding Author Email
12345678	Advances in Cancer Research	2023/10/15	John Doe	Pharma Inc.	john.doe@pharma.com
87654321	Novel Therapies for Cancer Treatment	2022	None	None	jane.smith@university.edu

4.3 Challenges

- **Email Extraction**: Emails are extracted from affiliation text using regex, which may not always correspond to the corresponding author.
- **Affiliation Filtering**: Some academic institutions may be misclassified as non-academic due to ambiguous keywords (e.g., "Research Institute").
- **API Limitations**: The PubMed API has rate limits, which can slow down data fetching for large queries.

---

5. Future Improvements

1. **Enhanced Email Extraction**:
   - Parse `<ContactInfo>` elements in the XML response to accurately identify corresponding author emails.
2. **Pagination Support**:
   - Add support for fetching more than 100 results by implementing pagination in the PubMed API requests.
3. **User Interface**:
   - Develop a graphical user interface (GUI) or web-based interface for easier interaction.

---

6. Conclusion

The PubMed Paper Fetcher is a functional and efficient tool for automating the process of fetching and analyzing research papers from PubMed. It provides valuable insights into authorship and affiliations, making it useful for researchers and analysts. With further improvements, the tool can become even more robust and user-friendly.

---

7. Appendix

7.1 Code Repository

The code for this project is available on GitHub:

GitHub Repository Link (https://github.com/yourusername/my_project)

7.2 Dependencies

- Python 3.8+
- `requests`
- `pytest` (for testing)
