Data Acquisition and Extraction:

Extraction and analysis of key data from academic and research paper on optimization

Kanupriya Jain, Nan An and Othman Hicheur

13/12/2024

Contents

- Project overview
- Process for fetching data using API and GROBID
- Processing files and saving them
- Database Creation
- Visualization of Database

Project overview

- Extracting key data from academic and research papers on optimization using arXiv and GROBID 0.7.2 like authors, keywords, citations ...
- Storage of the data in a structured database using sqlite3.
- Data cleaning
- Analysis of the data and visualization.

Process for fetching data using API and GROBID 1/3

- **Query and Fetch Metadata:** requests with arXiv API, extracting metadata (titles, authors, publication dates…) and PDF link.
- **Download paper:** download of the papers from arXiv.
- **Process with GROBID:** use of GROBID-generated TEI XML files to extract structured data.
- **Analyse Sections and References:** Parse TEI XML in order to identify the different part of the paper such as section, authors, publication years …

Process for fetching data using API and GROBID 2/3

Use of arXiv API to fetch metadata for papers matching a specified query:

- **Constructing the API Query:** query sent on arXiv API.
- **Parsing the XML Response:** parsing XML response and locate <entry> which represents a paper.
- **Extracting Metadata:** extract relevant fields from XML specific tags.
- **Handling PDF Links:** construct the PDF url by extracting article ID.
- **Storing Paper Data:** all extracted informations are stored in dictionary.
- **Changing the Date Format:** date format convert from %Y-%m-%dT%H:%M:%SZ (ISO 8601) to %B %d, %Y.

Process for fetching data using API and GROBID 3/3

Use of GROBID for extraction of contents and references:

- **Parsing TEI XML:** parsing of TEI XML generated by GROBID.
- **Extracting Sections:** sections are identified by , the content is extracted from <heads> tags, and the body text is aggregated from <p> tags.
- **Extracting References:** References are found with <biblStruct>, but not the first one (the paper itself). For each reference: title (<title>), authors (<authors>), publication year (<date>), url (extracting from tags if present).

Remark about citations:

One thing we can observed was that we were getting DOI as None for all the papers. We wanted to extract the citations data as well. We tried to use CrossRef or Semantic scholar. CrossRef API requires DOI get the data about citations and Semantic Scholar needs API Key and they have paused their service for some time so we were unable to extract this data.

Database Creation 1/3

Articles Table:

- id: Primary key
- title: String
- summary: Text
- published_date: Date
- updated_date: Date
- primary_category: String
- doi: String (nullable)
- link: String
- pdf_link: String (nullable)

Authors Table:

- id: Primary key
- name: String

Article_Authors Table:

- article_id: Foreign key to Articles
- author_id: Foreign key to Authors

Database Creation 2/3

Categories Table:

- id: Primary key
- term: String

Article_Categories Table:

- article_id: Foreign key to Articles
- category_id: Foreign key to Categories

Article_References Table:

- id: Primary key
- article_id: Foreign key to Articles (the citing paper)
- referenced_article_id: Foreign key to Articles (the cited paper, nullable if the cited paper isn't in the database)
- reference_title: String
- reference_doi: String (nullable)
- reference_link: String (nullable)

Database Creation 3/3

Keywords Table:

- id: Primary key
- keyword: String (e.g., "machine learning", "optimization")

Sections Table:

- id: Primary key
- article_id: Foreign key to Articles
- section_title: Title of the section (e.g., "Introduction")
- section_content: Full text of the section
- section_number: Order of the section in the document

Article_Keywords Table:

- id: Primary key
- article_id: Foreign key to Articles
- keyword_id: Foreign key to Keywords

Conclusion Table:

- id: Primary key
- article_id: Foreign key to the Articles table
- conclusion: Text content of the conclusion

Inserting Data

- Insert the metadata extracted from the API into the corresponding tables, such as the title, summary, links, into the Articles table.

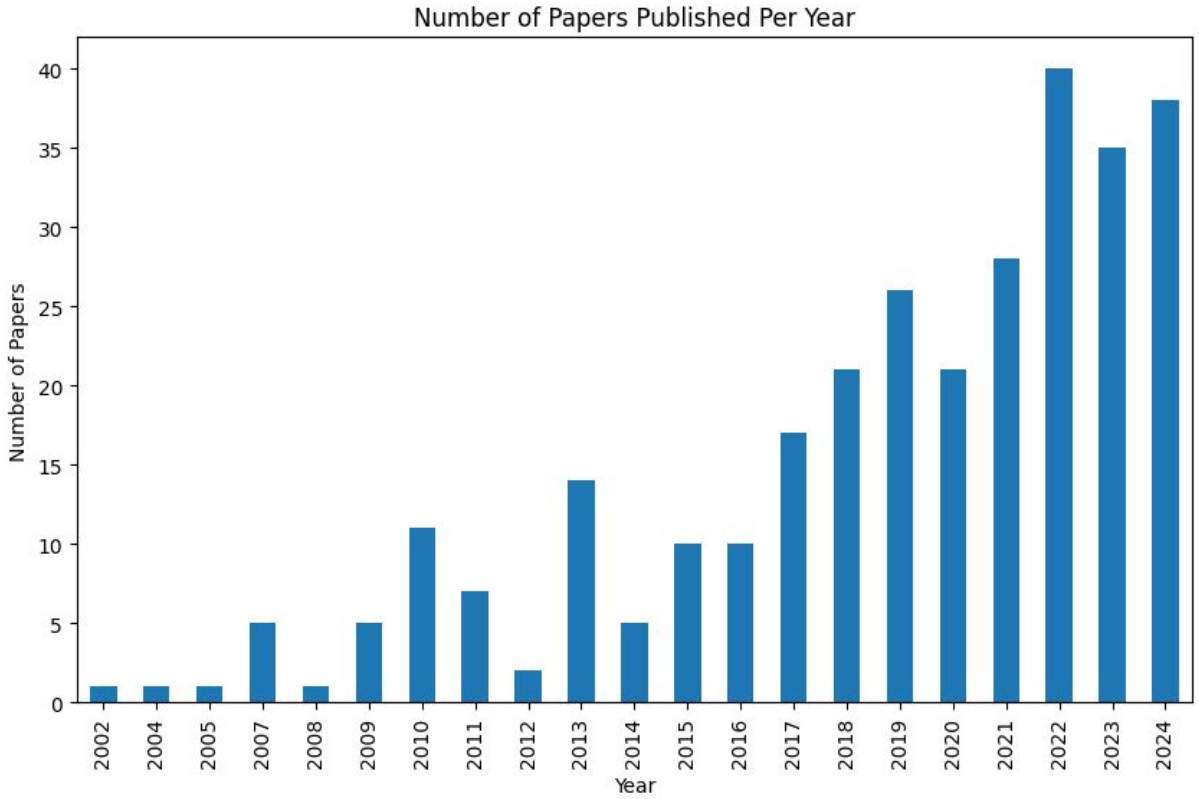
- Sections extracted from GROBID XML are added to the Sections table.

- Conclusions and keywords are stored separately in the Conclusion table and Keywords table, respectively.

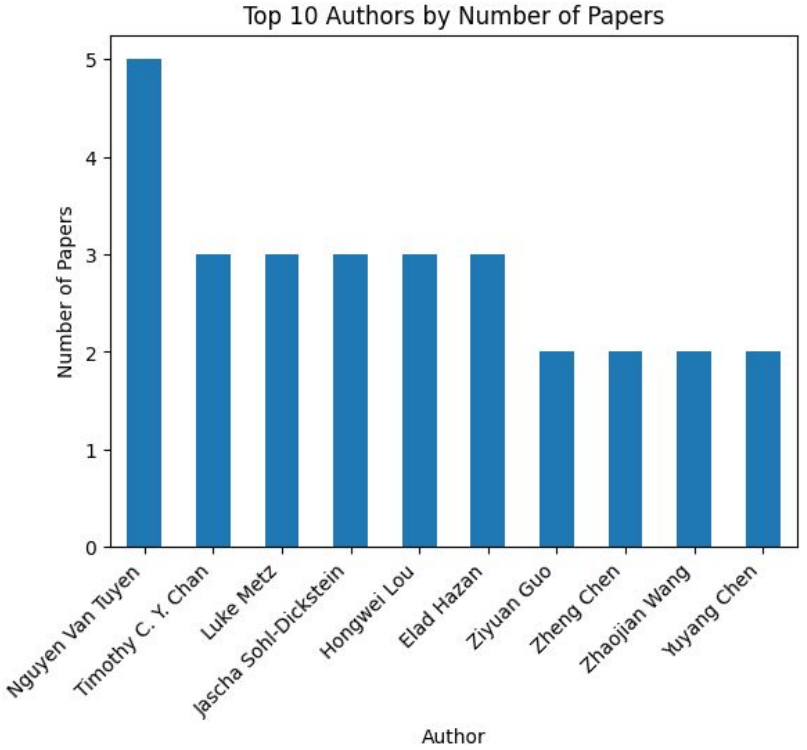
Data Cleaning

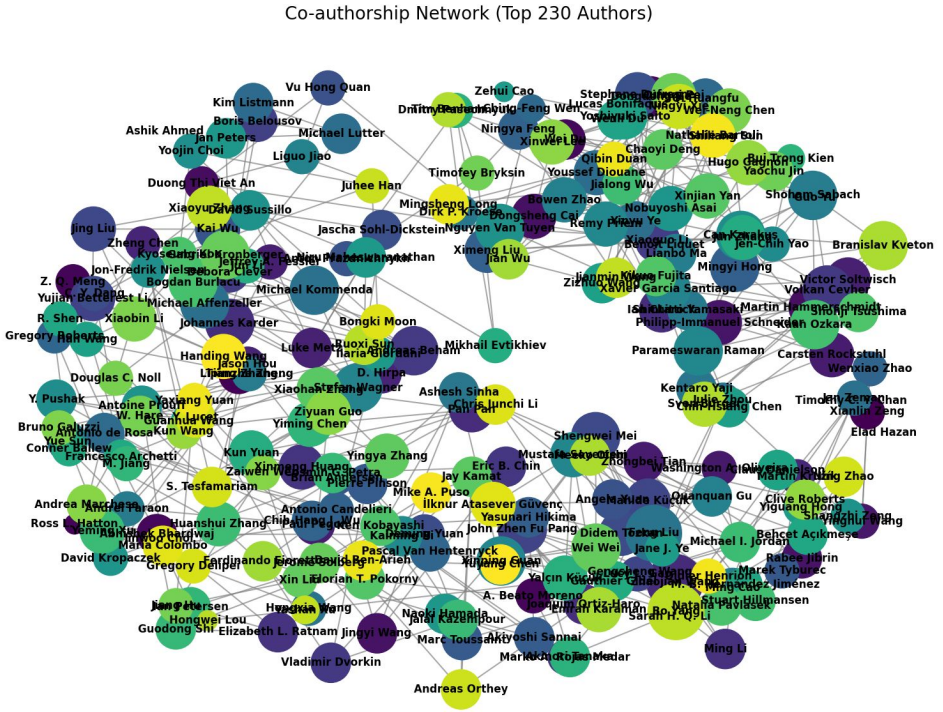
- Checked if `title` and published_date are NULL in the Articles table.
- We checked for missing references now in `Article_References`.
- Checked for duplicate `title` entries in the `Articles` table.
- Normalize the keyword by converting all keywords to lowercase.
- Checked for missing data in `sections` table as well.

Number of paper published each year

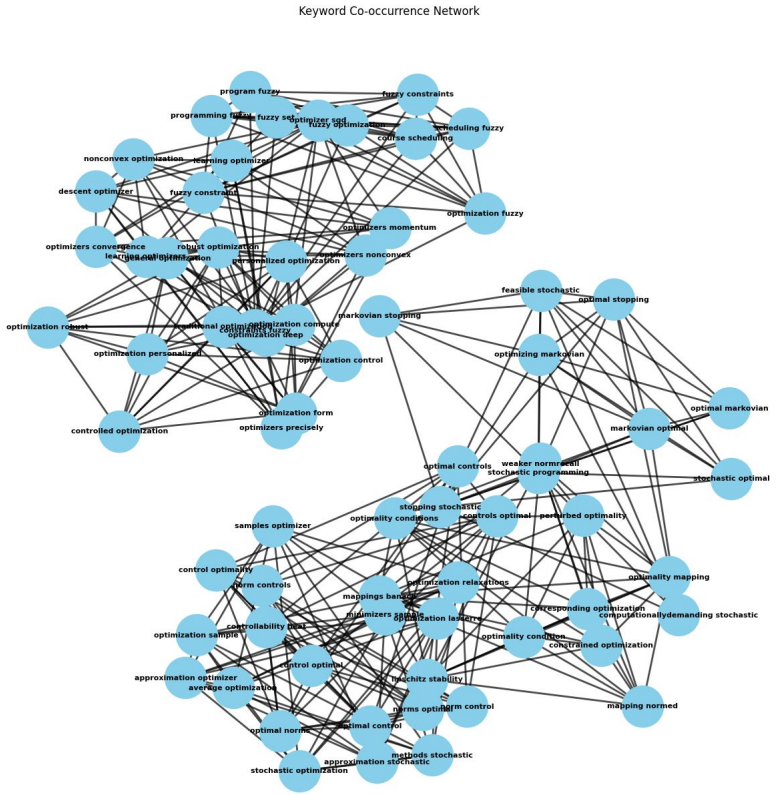


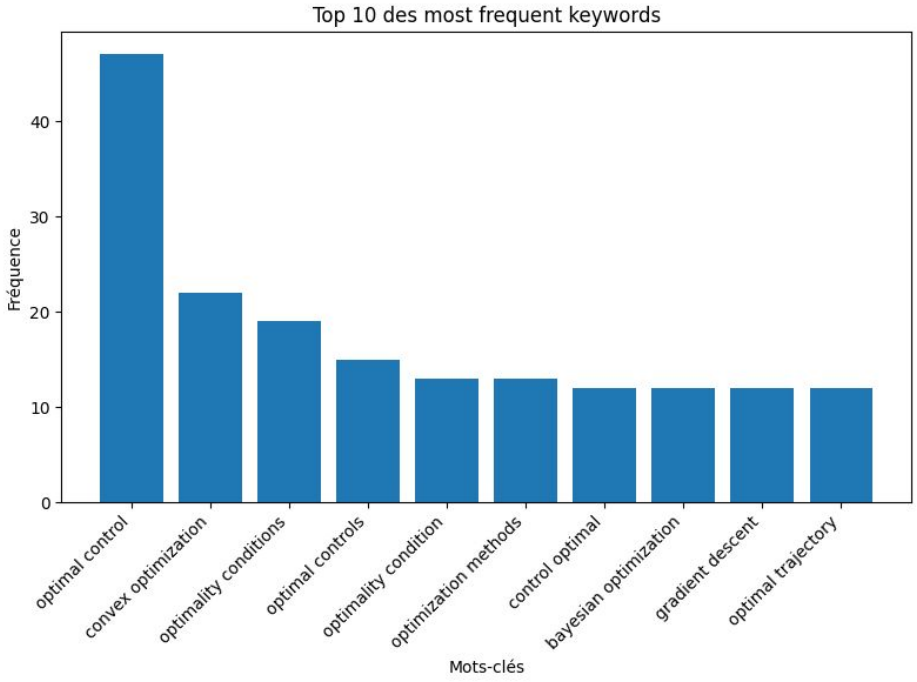
Top 10 Authors by number of papers and co-authorship network





Keywords graph for 600 words and most frequent keywords





Top 10 categories of papers