Development of an interactive tool for analyzing trends in medical research based on PubMed data

(Entwicklung eines interaktiven Werkzeugs zur Analyse von Trends in der medizinischen Forschung auf Basis von PubMed-Daten)

Pawel Wiezel

Hochschule Luzern - Informatik

CAS Data Engineering and Applied Data Science

Luzern, 15.02.2025

Table of Contents

Abstract

The expansion of research in the medical field makes it difficult to stay updated on specific topics and research trends, making trend analysis an essential tool. One of the primary sources for biomedical literature is PubMed, but it lacks built-in analytical tools. Fetching the necessary data from their database is a complex task. To address this challenge, this thesis focuses on developing an interactive tool for medical research trend analysis based on PubMed data. The tool, whose core is written in Python, uses the PubMed API to retrieve metadata from the PubMed database. It is based on topic-related tags (MeSH terms) and a time period defined by the user. The data is stored in a PostgreSQL database and used as a source for visual trend analysis in MS Power BI. The analysis takes the form of an interactive dashboard, which allows the user to customize the results by adjusting the filters in the visualizations. The tool provides a scalable approach for trend analysis and could be further developed to include more data sources, UI improvements, real-time updates and NLP enhancements.

Introduction

Motivation

The increasing number of scientific papers in biomedical sciences makes it difficult, if not impossible, to track specific topics and research trends. PubMed is a major database for biomedical literature and in 2023 alone, it published over 1.5 million papers [1]. Despite publishing large numbers of papers, PubMed lacks comprehensive built-in trend analysis tools, which would allow scientists and medical professionals to quickly analyze current or historical trends in science, because analyzing PubMed data manually is time-consuming and inefficient. Fortunately, to enable researchers to conduct such an analysis, PubMed offers a free API. However, using this API requires some coding knowledge, making it inaccessible to many users. An interactive tool that simplifies data fetching and storage while presenting results in a user-friendly dashboard could be useful for researchers without coding knowledge

Objective

The objective of this thesis is to develop a tool that enables users without coding knowledge to utilize PubMed’s API functionality and perform automated analyses on retrieved metadata, such as publication date, authors and MeSH terms (PubMed’s category tag). The core of the tool is written in

Python, allowing users to specify search terms and define the applicable time period for the query. The metadata is stored in an SQL database using a snowflake schema to ensure efficient processing. Data analysis is conducted in MS Power BI through an interactive dashboard, where users can apply flexible filters for customized data analysis and trends visualization.

Relevance and Contribution

The tool can support the researchers by providing an easy to use and time-saving platform for data analysis using PubMed metadata. It allows users to conduct trend identification through structured data processing. It establishes a foundation for future improvements, including easy scalability, NLP-based analysis, introduction of additional sources and real-time updates.

Background and Related Work

PubMed Database

PubMed is one of the biggest biomedical databases maintained by the National Center for Biotechnology Information (NCBI) which is a division of the National Library of Medicine (NLM). It provides an access to a vast repository of biomedical research literature. The core of PubMed's data is derived from MEDLINE, which is a database containing bibliographic information from numerous life science journals [2]. Each record in PubMed is structured in a standardized format, represented in XML language, to assure consistent data representation and retrieval. Key elements of a PubMed record include:

- PMID: A unique identifier assigned to each PubMed record.
- Title: The title of the published article.
- Abstract: A summary of the article's content.
- Authors: List of contributing authors.
- Journal Information: Details about the journal, including name, volume, issue and page numbers.
- Publication Date: The date when the article was published.
- MeSH Terms (Medical Subject Headings): Structured terms that categorize the article's content.

These elements are organized hierarchically within the XML structure, allowing for efficient parsing and data extraction [3].

Access to the PubMed database via API

To allow programmatic access to its databases, including PubMed, NCBI created a free API (Application Programming Interface) called Entrez Programming Utilities. It is a set of rules and protocols that allows different software applications to communicate with each other. It defines how requests and responses should be structured to access or manipulate data from another system [4]. Entrez is a search and retrieval system developed by the NCBI. It enables users to search and retrieve data from NCBI databases like PubMed. The E-utilities are a set of server-side programs that provide an interface into the Entrez system, which operate via HTTP requests, enabling automated searches and retrieval of PubMed data. It contains various tools designed for specific tasks including:

- ESearch: Searches the texts in Entrez databases and retrieves a list of PMIDs (PubMed identifiers) for articles that match the searched term.
- EFetch: Retrieves detailed metadata (title, authors, journal, publication date) for the found articles.
- ESummary: Provides document summaries for a list of PMIDs.
- ELink: Retrieves related articles and links between PubMed and other NCBI databases.

NCBI enforces request limits (3 requests per second) to prevent excessive traffic. Using an API key increases the limit to 10 requests per second. Private API key can be generated after opening an account on NCBI page (ncbi.nlm.nih.gov/account/t) [5] [6].

Other examples of existing APIs for Literature Retrieval

There are other APIs, which provide access to scientific literature, allowing researchers to retrieve data about scientific article. These APIs vary in terms of the scope of the data they cover, access restrictions and functionality. For this thesis PubMed API was chosen, because it focuses on medical literature. Examples of other popular APIs for scientific literature include:

Elsevier Scopus API

- Provides access to Scopus - a large database of peer-reviewed literature.

- Contains citation data, author metrics and journal impact scores.
- Requires an institutional subscription for full access.
- requires Elsevier subscription [7].

Springer Nature API

- Allows searching in Springer journals, books and conference papers.
- Provides full-text retrieval for open-access content.
- To retrieve articles, which are not open access, a paid Springer subscription is required [8].

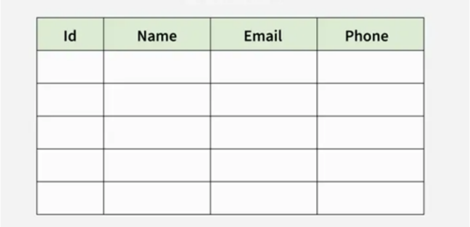
Data storage

Database

The data fetched by the API requires efficient storage and retrieval mechanisms. One storage possibility is a database. A structured database ensures that PubMed metadata is organized, accessible and ready for analysis. It allows for efficient querying, filtering and data analysis. A well-designed schema helps to store metadata efficiently avoiding redundancy. A database must have a schema, which define the structure, organization and relationships of data in a database system, ensures efficient storage, retrieval and integrity of data. There are various types of schemas, including:

Flat Model

- Simplest type of schema.
- Data is stored in a single table.
- Useful for small datasets with minimal complexity.
- Limitations:
    - No relationships between records.
    - Data redundancy increases with size.



Id	Name	Email	Phone

Figure 1: Flat Schema [9]

Relational Schema

- Organizes data into multiple tables, which are related with each other using primary keys and foreign keys.
- Advantages:
    - Relationships between data are strictly enforced, what ensures consistency.
    - Updates are flexible, because updating a single table updates all related records.
    - The Data is divided into smaller related tables to avoid duplication.
- Limitations:
    - Requires complex joins to retrieve complete datasets, which can be inefficient for large queries.

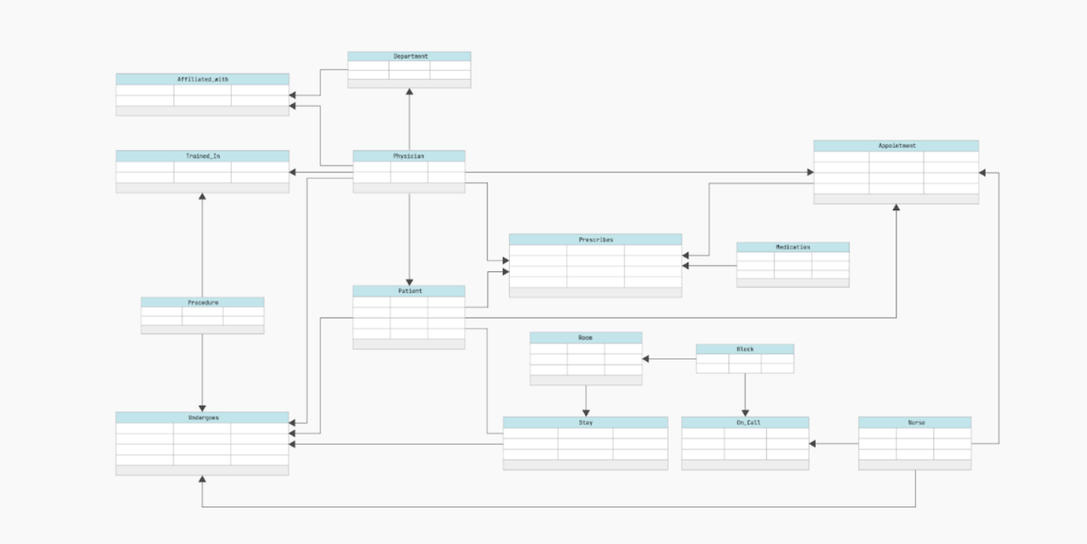


Figure 2: Relational Schema [10]

Hierarchical Model

- Data is structured in a tree-like format with parent-child relationships.
- Each parent can have multiple child records, but each child has only one parent.
- Used in file systems, XML databases and legacy applications.
- Limitations:
    - Not flexible for complex relationships.
    - Querying across multiple levels can be inefficient.

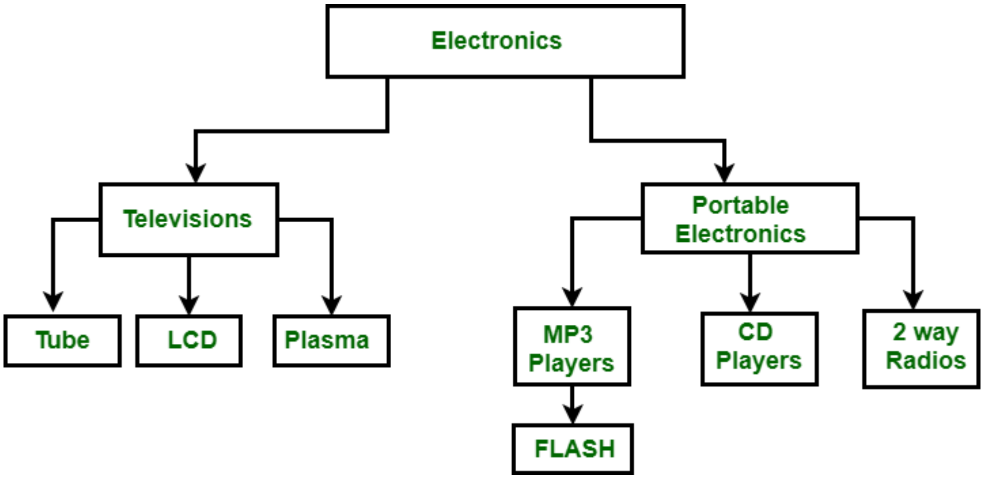


Figure 3: Hierarchical schema [9]

Star Schema

- Commonly used in data warehousing and business intelligence.
- Central (fact) table stores numerical data (like article ID).
- Satellite (dimension) tables contain descriptive information (publication date, authors).
- Advantages:
  - Reduced number of tables and joins by storing redundant data in dimension tables enables faster query performance.
  - Easy to understand and implement.
- Limitations:
  - Dimension tables can store repeated values, what can cause data redundancy, increase storage requirements and costs.
  - Updates must be executed in multiple tables, risking inconsistent data structure.
  - Many-to-many relationships require additional linking tables.

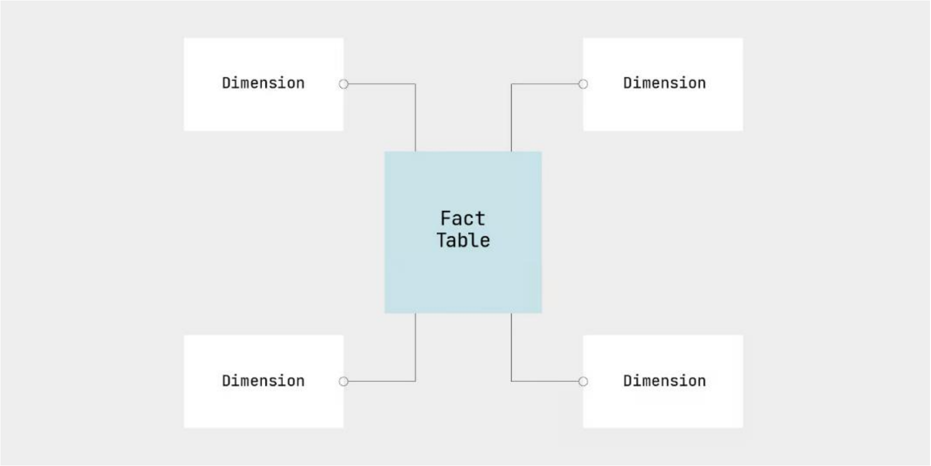


Figure 4: Star schema [10]

Snowflake Schema

- An extension of the star schema with further normalization.
- Dimension tables are split into sub-dimensions, reducing data redundancy.
- Advantages:
  - Saves storage space by avoiding duplication.
  - Useful when dimension tables contain hierarchical relationships.
- Limitations:
  - Slower performance compared to the star schema, because of increased complexity [11] [12] [13] [10].

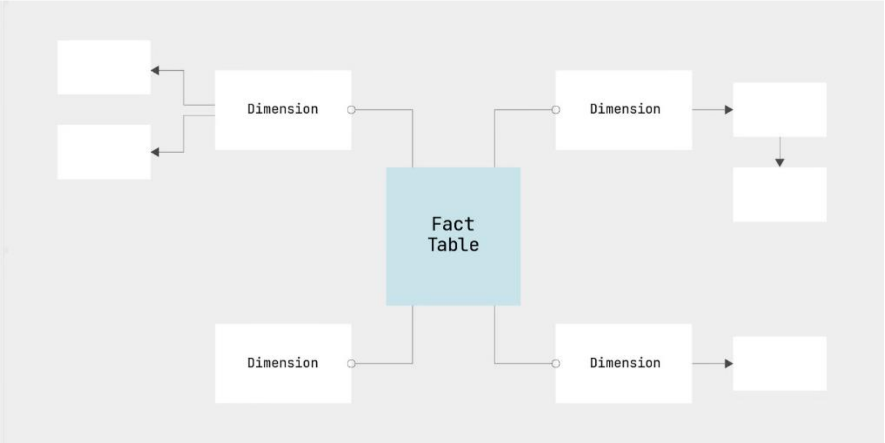


Figure 5: Snowflake schema [10]

ETL and Data Pipelines

Definition of ETL and Data Pipeline

- ETL (Extract, Transform, Load) - a process used to extract data from sources, transform it into a structured format and load it into a database. It is a core component of data warehousing. ETL is a traditional variation of the process, where. data is transformed before loading into the database. Another, modern approach is ELT, where data is first loaded in a raw form into a data storage, then transformed inside the target system as needed.

- Data pipeline - A data pipeline is a system that moves data from a source to a destination, automating processes such as collection, transformation and storage. It ensures efficient data flow, enabling integration, analysis and usage in applications.

Existing ETL Approaches in Literature Retrieval

- ETL is widely used in scientific data processing, including:
  - Bioinformatics and Healthcare Analytics – Extracting and structuring patient records.
  - Literature Analysis – Automating retrieval and storage of research articles from databases like PubMed or Scopus.
  - Trend Detection – Processing metadata to identify emerging research topics [14] [15] [16].

Relevant Tools:

Various tools are available for database management and visualization. For the articles' fetching Python was preferred over R due to its extensive libraries for API handling and data manipulation. Python code was developed using MS VS Code. MySQL, PostgreSQL and SQLite are common relational database systems. PostgreSQL was chosen due to its scalability and better performance while handling large datasets. A snowflake schema was chosen for its ability to efficiently manage many-to-many relationships between articles, authors and search terms. It seamlessly integrates with Power BI, allowing structured data analysis in an interactive dashboard. Additionally, its scalable design allows new facts and dimensions to be incorporated without requiring significant structural modifications.

Power BI was selected due to its seamless integration with SQL databases, interactive dashboards and lower cost compared to paid alternatives like Tableau [17] [18] [19].

Methodology

Data Collection

The tool developed in this project retrieves biomedical research metadata from the PubMed API (Entrez API), which provides access to numerous scientific publications. The core of the tool is Python script, which automates data extraction, using the requests library to query the API. The collected metadata includes article titles, authors, journals, publication dates and MeSH terms. The user specifies a publication date range and search terms. The code then matches the provided search terms with official MeSH terms and uses the corresponding ones to retrieve relevant articles. Search term is skipped if does not match any MeSH term. Search terms list was introduced to differentiate between official MeSH terms and a list with terms specified by user.

OpenAI's ChatGPT 4o and 3o-mini-high models were used for debugging and improving certain sections of the code to ensure efficiency.

Data Processing and Transformation

Data Extraction and Format Conversion

Articles are retrieved from PubMed API using Python (requests library) in XML format. The XML response is parsed using xml.etree.ElementTree, extracting key metadata fields (PMID, authors, title, journal, publication date and MeSH terms). Data is structured into a Pandas DataFrame for further processing.

Data Cleaning and Standardization

Missing values (e.g., missing author names or publication dates) are handled by setting defaults or excluding incomplete records. Duplicate records are removed based on PMID uniqueness. To standardize the data search terms are converted to lowercase to prevent duplicates and date format is normalized to YYYY-MM-DD for consistency. The file is then saved in parquet format to maximize performance.

Data Storage

A PostgreSQL database stores the transformed data, structured using a snowflake schema with a many-to-many relationship between articles and search terms. The schema consists of:

- Fact Tables:
    - fact_articles – Stores article metadata, linking to titles, journals and authors.
    - bridge_articles_search_terms – A bridge table providing the many-to-many relationship between articles and search terms.
    - bridge_articles _authors – A bridge table providing the many-to-many relationship between articles and authors.
- Dimension Tables:
    - dim_titles – Unique titles of articles.
    - dim_journals – Unique journal names.
    - dim_authors – Unique author names.
    - dim_search_terms – Unique search terms used in PubMed queries.

The snowflake schema structure provides fast query performance by minimizing joins while maintaining data consistency at the same time. It also allows fast query execution by reducing the number of joins required for analytical queries. The bridge tables enable many-to-many relationships, ensuring accurate mapping of articles to multiple search terms and articles to multiple authors. Data is inserted into PostgreSQL using Psycopg2, ensuring structured and scalable storage for further processing in Power BI.

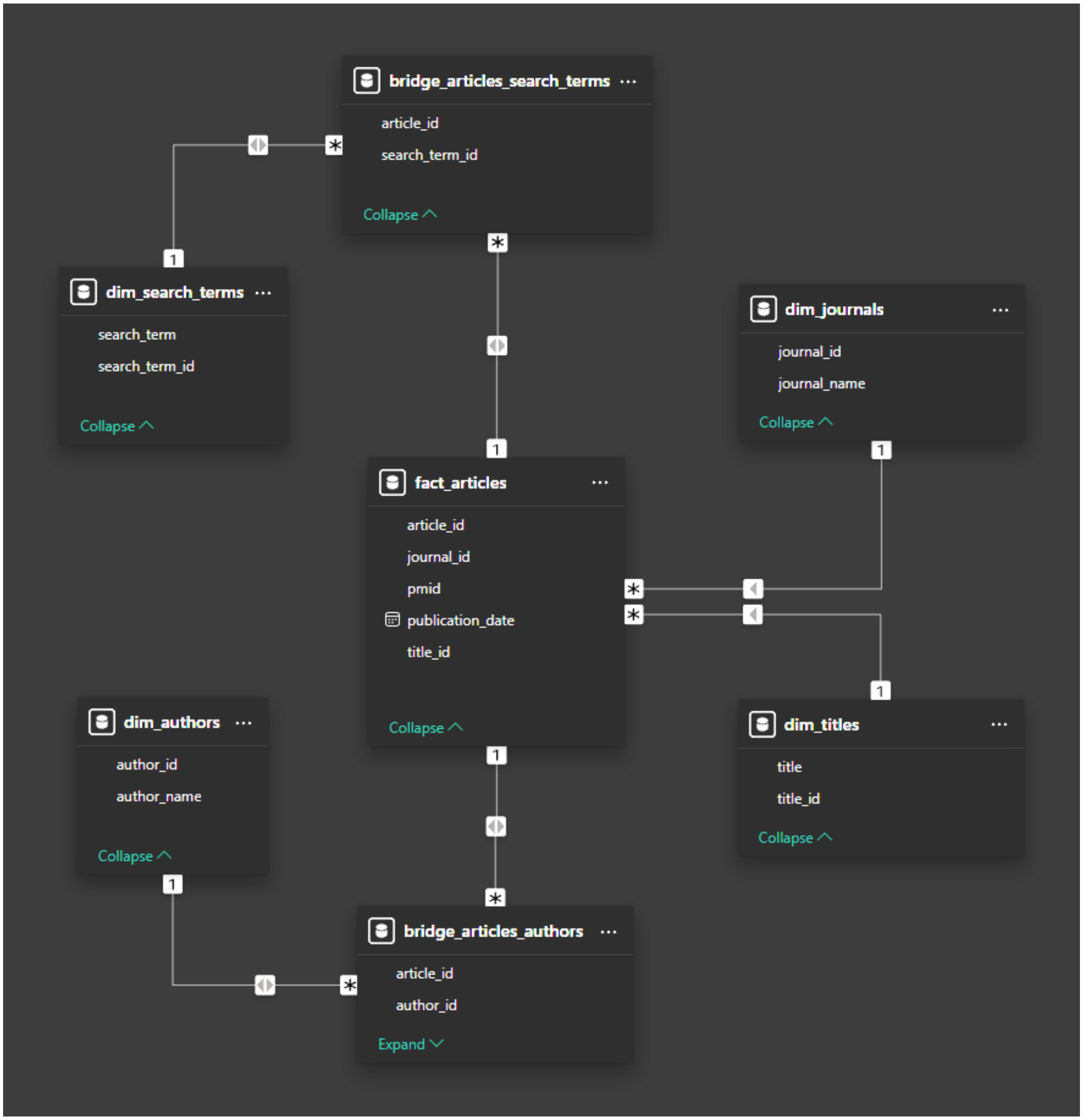


Figure 6: Database Schema used in the project – Screenshot from MS Power Bi.

Data Analysis and Visualization

Processed data is loaded into Power BI, where interactive dashboards display trends in MeSH term frequency, publication volume and author contributions. The snowflake schema enables fast filtering and drill-down analysis, improving data exploration.

A flowchart presented on Figure 7: Pipeline for processing PubMed data, from user input to trend analysis in Power BI. shows the complete pipeline, from user input to the data visualization.

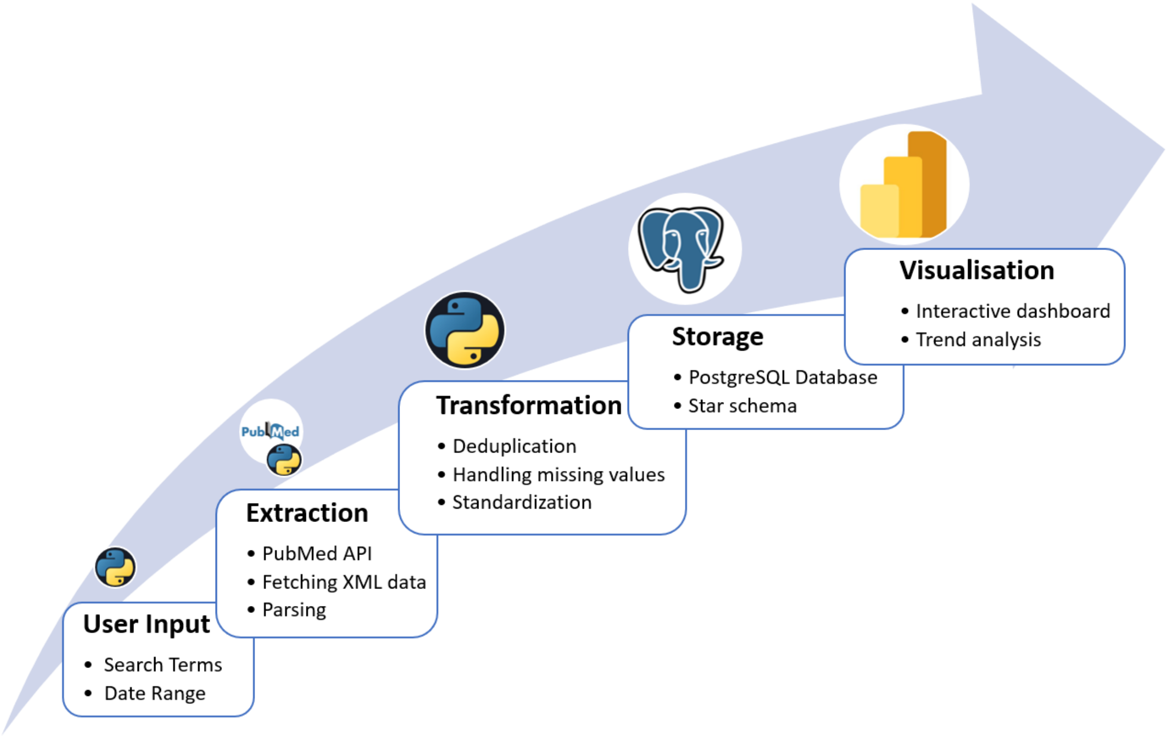


Figure 7: Pipeline for processing PubMed data, from user input to trend analysis in Power BI.

Automation

An automated ETL pipeline ensures that data retrieval, transformation and loading occur without manual intervention. The database schema supports scalability, allowing the addition of new dimensions and facts with minimal structural changes.

Results and Discussion

This chapter presents the outcomes of the ETL pipeline, verifying the functionality and discussing its feasibility for analyzing trends in medical research articles. The goal is to show, that data is successfully retrieved, processed and stored in PostgreSQL and that Power BI is capable of meaningful trend analysis. For validation, a set of predefined search terms—"medical genetics", "radiology", "cardiology", "dermatology", "oncology", "neurology", "artificial intelligence" and "machine learning"—was selected, with the publication date range restricted to 01.01.2024 to 31.12.2024. The SQL queries presented in this chapter were executed using PostgreSQL within the pgAdmin 4 environment.

Data Retrieval

Python code using the PubMed API retrieved the data based on user-defined search terms and a specified date range. The retrieval process successfully returned structured XML data, which was then parsed and transformed into a structured format. The extracted metadata, including titles, journals, authors and publication dates, was standardized to ensure consistency. The implemented loop iterates over a list of search terms, matching them with MeSH terms and querying the PubMed database for relevant articles. For every term, a request is sent to the PubMed E-Utilities API using a predefined query structure that includes MeSH terms and a publication date filter. For each search term, a query is formulated and passed as a parameter in the API request. The Entrez search (esearch.fcgi) endpoint retrieves article identifiers (PMIDs) and the fetch (efetch.fcgi) endpoint is used to extract metadata (such as title, authors, journal and publication date). If the total number of results exceeds a predefined limit of records per request (10000), the script will implement pagination, ensuring that all available data is retrieved in batches. The retrieved XML response is parsed, extracting key metadata fields. Each article's information is stored in a structured format, ensuring that multiple authors and search terms are properly linked to their corresponding articles. Data is appended in long format, where each author is stored as a separate row to facilitate relational database storage and analysis. Once all search terms have been processed, the collected data is consolidated into a Pandas DataFrame, ensuring efficient handling and further processing within the data pipeline.

The transformed data was successfully loaded into a PostgreSQL database using a snowflake schema, enabling efficient querying and trend analysis. A many-to-many relationship structure was implemented to associate articles with multiple search terms, allowing for more flexible data exploration.

To ensure, that the data is correctly fetched from the source and transformed into a structured format before being stored multiple tests were performed. The results of these tests provided confidence that the extracted dataset is reliable and can be used for further analysis. The tests were conducted directly on the SQL server using pgAdmin 4. The dataset obtained after fetching the data the contained 28773 unique articles, confirming the integrity of the retrieval pipeline. The verification process ensured that no duplicate records were present and that every article was correctly linked to at least one search term. Additionally, a summary table was generated to verify the distribution of articles across search terms, confirming that the MeSH terms assignments were correctly structured and no search terms were missing expected associations.

Verification:

- Total number of retrieved articles:

Query 1:

```
SELECT COUNT(*) AS total_articles FROM fact_articles;
```

Result:

Table 1: Table showing the number of articles retrieved from PubMed's database.

total_articles
28773

- Confirming number of Unique PMIDs to ensure no duplicates. Requires empty table to pass:

Query 2:

```
SELECT pmid, COUNT(*)
FROM fact_articles
GROUP BY pmid
HAVING COUNT(*) > 1;
```

Result:

count = 0, table confirming, that there are no duplicates in the database.

- Ensuring that every article in fact_articles is linked to at least one search term by identifying articles with no match in bridge_articles_search_terms. The result must be an empty table to pass the test:

Query 3:

```
SELECT f.pmid, COUNT(b.search_term_id) AS term_count
FROM fact_articles f
LEFT JOIN bridge_articles_search_terms b ON f.article_id = b.article_id
GROUP BY f.pmid
HAVING COUNT(b.search_term_id) = 0;
```

Result:

term_count = 0, table confirming, that there are no articles, which are not linked to any search terms in the bridge table.

- Verifying, that all articles are correctly linked to search terms and identifies potential missing or incorrect assignments:

Query 4:

```
SELECT s.search_term, COUNT(*) AS article_count
FROM bridge_articles_search_terms b
JOIN dim_search_terms s
ON b.search_term_id = s.search_term_id
GROUP BY s.search_term
ORDER BY article_count DESC;
```

Result:

Table 2: Number of articles associated with each search term.

"search_term"	"article_count"
"machine learning"	8864
"artificial intelligence"	8444
"radiology"	8407
"oncology"	7486
"cardiology"	826
"dermatology"	717
"neurology"	463
"medical genetics"	397

Database Validation

The fact_articles table correctly references dim_titles and dim_journals, ensuring proper classification of articles within the database. The bridge tables (bridge_articles_authors and bridge_articles_search_terms) establish many-to-many relationships, linking articles to authors and search terms. This structure ensures data integrity and enables efficient querying.

A verification query confirmed that all foreign key constraints work correctly, preventing missing references and keeping the data consistent. The validation process ensured that all articles in fact_articles reference valid titles (dim_titles) and journals (dim_journals), with all queries returning an empty result, confirming integrity. The validation also confirmed that all records have valid publication dates within the expected range. Additionally, a sample of records was retrieved to verify that fact_articles correctly links to its dimension tables, demonstrating that articles are accurately classified and their relationships are properly maintained within the database.

Verification:

- Verifying, that all articles in fact_articles reference valid titles and journals. The result must be an empty table to pass the test:

Query 5:

```
SELECT COUNT(*) AS missing_titles
FROM fact_articles f
LEFT JOIN dim_titles t ON f.title_id = t.title_id
WHERE t.title_id IS NULL;

SELECT COUNT(*) AS missing_journals
FROM fact_articles f
LEFT JOIN dim_journals j ON f.journal_id = j.journal_id
WHERE j.journal_id IS NULL;

SELECT f.pmid
FROM fact_articles f
LEFT JOIN bridge_articles_authors b ON f.article_id = b.article_id
WHERE b.article_id IS NULL;

SELECT f.pmid
FROM fact_articles f
LEFT JOIN bridge_articles_search_terms b ON f.article_id = b.article_id
WHERE b.article_id IS NULL;
```

Results:

Each query returned an empty table, confirming that all articles reference valid titles and journals.

- Check for missing or invalid dates. The result must be an empty table to pass the test:

Query 6:

```
SELECT COUNT(*) AS invalid_dates
FROM fact_articles
WHERE publication_date IS NULL
OR publication_date < '1900-01-01'
OR publication_date > CURRENT_DATE;
```

Result:

invalid_dates = 0, confirming all records contain valid publication dates.

- Sample records demonstrating correct relationships between fact and dimension tables:

Query 7:

```
SELECT f.pmid, t.title, j.journal_name, f.publication_date
FROM fact_articles f
JOIN dim_titles t ON f.title_id = t.title_id
JOIN dim_journals j ON f.journal_id = j.journal_id
LIMIT 10;
```

Result:

Table 3: Sample records demonstrating correct relationships between fact and dimension tables.

pmid	title	journal_name	publication_date
39839483	Family planning and preimplantation testing: family experiences in congenital adrenal hyperplasia.	Frontiers in endocrinology	2024-01-01
39766813	Clinical and Cytogenetic Impact of Maternal Balanced Double Translocation: A Familial Case of 15q11.2 Microduplication and Microdeletion Syndromes with Genetic Counselling Implications.	Genes	2024-11-29
39766768	Targeted Genetic Education in Dentistry in the Era of Genomics.	Genes	2024-11-22
39715138	Medical genetics as a basis for personalized medicine in contemporary Ukraine.	Wiadomosci lekarskie (Warsaw, Poland : 1960)	2024-01-01
39616141	Recurrence Risks in Congenital Anomalies: A Comprehensive Guide for Parental Counseling.	NeoReviews	2024-12-01
39602801	Preliminary Screening for Hereditary Breast and Ovarian Cancer Using an AI Chatbot as a Genetic Counselor: Clinical Study.	Journal of medical Internet research	2024-11-27
39575078	Medical Genetics for the Undiagnosed and Rare Patient: "Chasing Zebras".	Missouri medicine	2024-07-01
39575065	Genetics and Primary Care: Raising Awareness and Enhancing Cooperation.	Missouri medicine	2024-07-01
39571977	[Introduction of Online Genetic Counseling for Hereditary Tumors Using an Online Medical Consultation Application].	Gan to kagaku ryoho. Cancer & chemotherapy	2024-10-01
39565467	Cascade genetic testing in hereditary cancer: exploring the boundaries of the Italian legal framework.	Familial cancer	2024-11-20

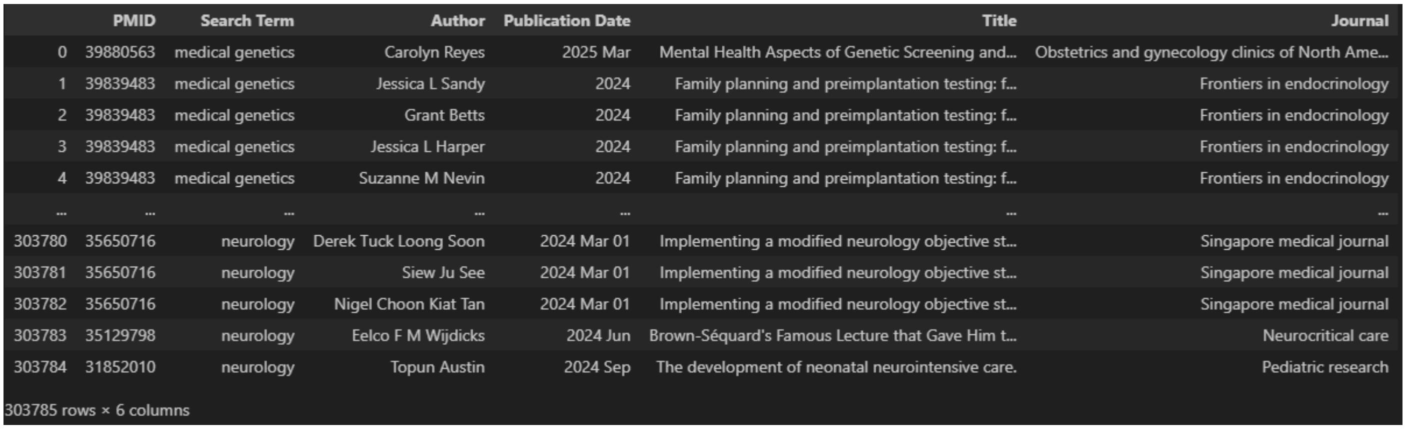
Data Transformation and Cleaning

The cleaning process ensured that the extracted data was structured and usable for analysis. Articles missing journal and author names were assigned *"Unknown" (*“Unknown” when there is no author tag metadata in fetched XML file, “Unknown Unknown” when first and last name information is missing and if only first or only last name is missing, only the missing part will be set as “Unknown”). This approach maintained the integrity of author attributions while preventing incomplete records from affecting analysis.

Duplicate PMIDs were removed to ensure that each article appears only once in the dataset and therefore to prevent duplicate articles from distorting trend analysis. Articles with missing PMID number were deleted for credibility reasons.

To standardize the data, search terms were converted to lowercase to prevent duplication due to case variations (e.g., *"Radiology”* vs. *"radiology"*) avoiding misclassification in MeSH term-based analysis. Additionally, publication dates were reformatted to the YYYY-MM-DD format for consistency and articles with missing dates were dropped to ensure completeness in timeline-based analyses.

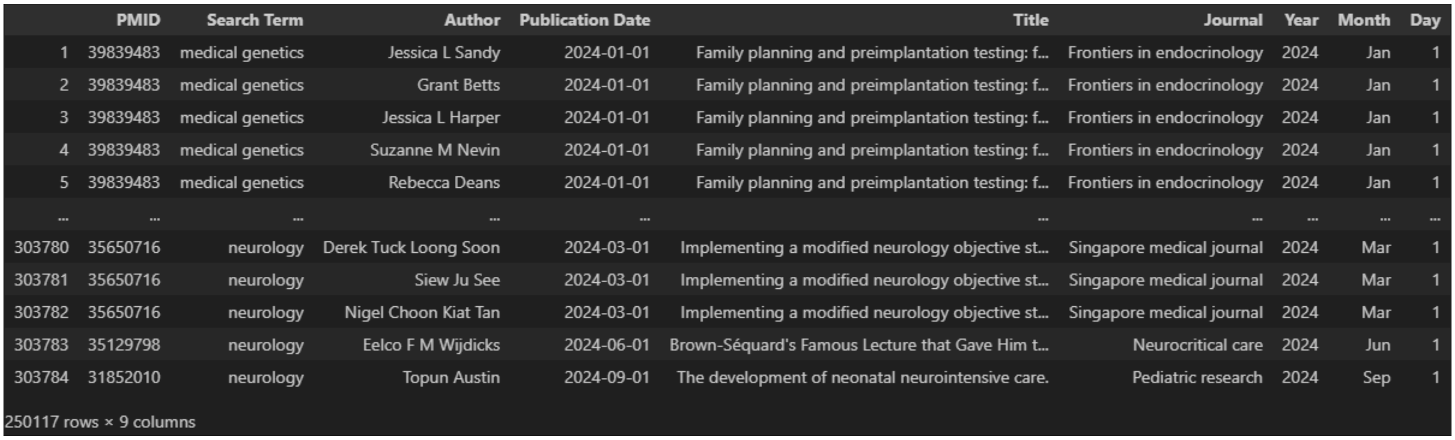
A detailed breakdown of the data cleaning process, including SQL queries and Python scripts, is available in the attached Jupyter Notebook files.



	PMID	Search Term	Author	Publication Date	Title	Journal
0	39880563	medical genetics	Carolyn Reyes	2025 Mar	Mental Health Aspects of Genetic Screening and...	Obstetrics and gynecology clinics of North Ame...
1	39839483	medical genetics	Jessica L Sandy	2024	Family planning and preimplantation testing: f...	Frontiers in endocrinology
2	39839483	medical genetics	Grant Betts	2024	Family planning and preimplantation testing: f...	Frontiers in endocrinology
3	39839483	medical genetics	Jessica L Harper	2024	Family planning and preimplantation testing: f...	Frontiers in endocrinology
4	39839483	medical genetics	Suzanne M Nevin	2024	Family planning and preimplantation testing: f...	Frontiers in endocrinology
...	...	...	...	...	...	...
303780	35650716	neurology	Derek Tuck Loong Soon	2024 Mar 01	Implementing a modified neurology objective st...	Singapore medical journal
303781	35650716	neurology	Siew Ju See	2024 Mar 01	Implementing a modified neurology objective st...	Singapore medical journal
303782	35650716	neurology	Nigel Choon Kiat Tan	2024 Mar 01	Implementing a modified neurology objective st...	Singapore medical journal
303783	35129798	neurology	Eelco F M Wijdicks	2024 Jun	Brown-Séquard's Famous Lecture that Gave Him t...	Neurocritical care
303784	31852010	neurology	Topun Austin	2024 Sep	The development of neonatal neurointensive care.	Pediatric research

303785 rows × 6 columns

Figure 8: Pandas DataFrame before cleaning in Visual Studio Code.



```
          PMID   Search Term               Author  Publication Date                                          Title                     Journal  Year  Month  Day
     1  39839483  medical genetics        Jessica L Sandy      2024-01-01   Family planning and preimplantation testing: f...   Frontiers in endocrinology  2024    Jan    1
     2  39839483  medical genetics            Grant Betts      2024-01-01   Family planning and preimplantation testing: f...   Frontiers in endocrinology  2024    Jan    1
     3  39839483  medical genetics       Jessica L Harper      2024-01-01   Family planning and preimplantation testing: f...   Frontiers in endocrinology  2024    Jan    1
     4  39839483  medical genetics       Suzanne M Nevin      2024-01-01   Family planning and preimplantation testing: f...   Frontiers in endocrinology  2024    Jan    1
     5  39839483  medical genetics         Rebecca Deans      2024-01-01   Family planning and preimplantation testing: f...   Frontiers in endocrinology  2024    Jan    1
   ...       ...               ...                   ...             ...                                                ...                          ...   ...    ...  ...
303780  35650716         neurology  Derek Tuck Loong Soon      2024-03-01  Implementing a modified neurology objective st...     Singapore medical journal  2024    Mar    1
303781  35650716         neurology            Siew Ju See      2024-03-01  Implementing a modified neurology objective st...     Singapore medical journal  2024    Mar    1
303782  35650716         neurology    Nigel Choon Kiat Tan      2024-03-01  Implementing a modified neurology objective st...     Singapore medical journal  2024    Mar    1
303783  35129798         neurology      Eelco F M Wijdicks      2024-06-01  Brown-Séquard's Famous Lecture that Gave Him t...          Neurocritical care  2024    Jun    1
303784  31852010         neurology          Topun Austin      2024-09-01     The development of neonatal neurointensive care.          Pediatric research  2024    Sep    1
250117 rows × 9 columns
```

Figure 9: Dataframe after cleaning in MS Visual Studio Code.

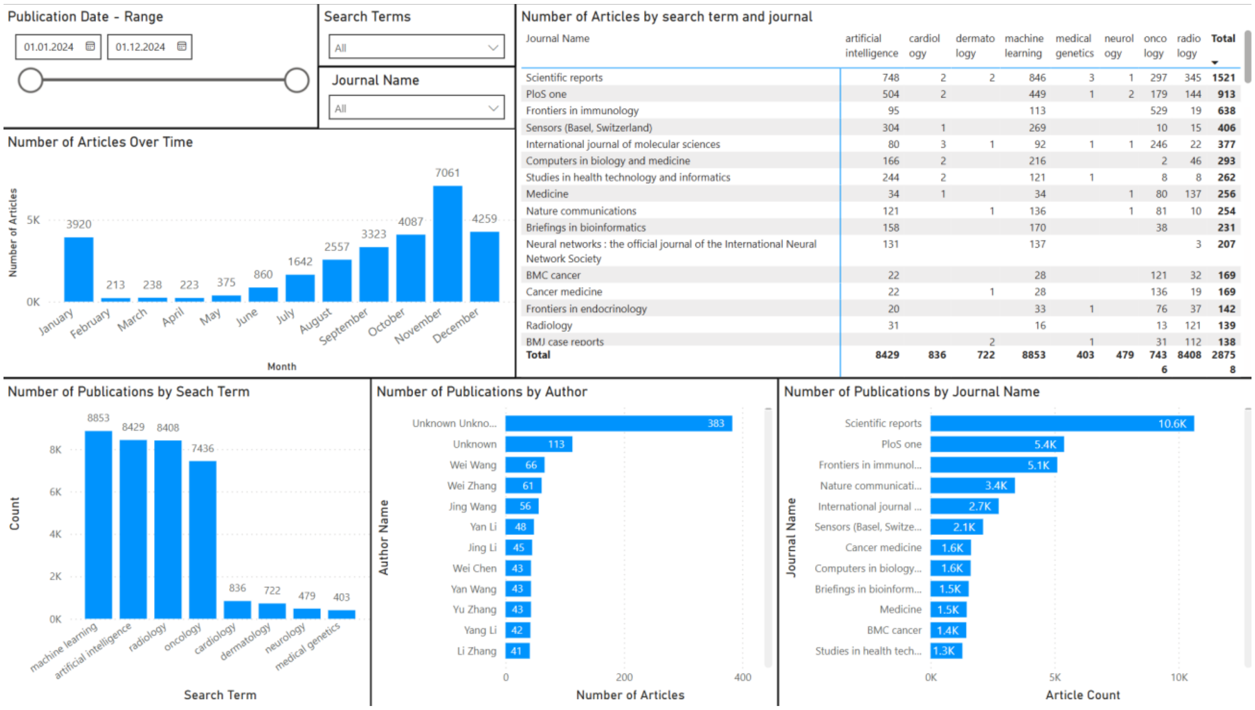
Power BI Visualization Output

The processed data was successfully loaded from the PostgreSQL database into Power BI, where an example of interactive dashboard with was created to perform data analysis. The dashboard serves as an example of the project's feasibility, showcasing how the processed data can be dynamically explored through multiple analytical perspectives. The dashboard contains slicers for publication date, search terms and journal names. These filters allow users to dynamically refine the visualizations to specific criteria, enabling a customized exploration of the dataset. While visualizations demonstrate the potential of the system, the primary focus of this project is the data pipeline, rather than the visualization itself. The dashboard contains following visualizations:

- Number of Articles Over Time: Displays the distribution of articles published across months in 2024. The data highlights peak publication periods, such as December and January, which observed a significant rise in the number of articles.
- Number of Publications by Search Term: This visualization identifies the most common search terms in the dataset. Terms like "machine learning," "artificial intelligence", "radiology" and “oncology” dominate, demonstrating their relevance in medical research trends.
- Number of Publications by Author: Identifies the most active authors within the dataset, such as Wei Wang and Wei Zhang, who each contributed a significant number of articles. Many author metadata was missing, which is depicted as “Unknown” in place of authors name and/or surname or if an author tag was missing completely.
- Number of Publications by Journal Name: Highlights the journals contributing the most articles. For example, *Scientific Reports* and *PLOS One* lead in publication volume, making them central contributors to the dataset.

- Number of Articles by Search Term and Journal: A matrix view combining search terms and journal names provides detailed insights into which journals are publishing articles on specific topics, giving the user a quick overview about the numbers.

The dashboard demonstrates the potential of the processed data for analyzing publication trends, identifying most active contributors and exploring topic-specific research outputs, further confirming the feasibility of the project.



Publication Date - Range

01.01.2024 01.12.2024

Search Terms

All

Journal Name

All

Number of Articles by search term and journal

Journal Name	artificial intelligence	cardiology	dermatology	machine learning	medical genetics	neurology	oncology	radiology	Total
Scientific reports	748	2	2	846	3	1	297	345	**1521**
PloS one	504	2		449	1	2	179	144	**913**
Frontiers in immunology	95			113			529	19	**638**
Sensors (Basel, Switzerland)	304	1		269			10	15	**406**
International journal of molecular sciences	80	3	1	92	1	1	246	22	**377**
Computers in biology and medicine	166	2		216			2	46	**293**
Studies in health technology and informatics	244	2		121	1		8	8	**262**
Medicine	34	1		34		1	80	137	**256**
Nature communications	121		1	136		1	81	10	**254**
Briefings in bioinformatics	158			170			38		**231**
Neural networks : the official journal of the International Neural Network Society	131			137				3	**207**
BMC cancer	22			28			121	32	**169**
Cancer medicine	22		1	28			136	19	**169**
Frontiers in endocrinology	20			33	1		76	37	**142**
Radiology	31			16			13	121	**139**
BMJ case reports			2		1		31	112	**138**
Total	**8429**	**836**	**722**	**8853**	**403**	**479**	**7436**	**8408**	**28758**

Figure 10: Proposed dashboard in MS Power Bi.

Discussion: System Feasibility and Limitations

The ETL pipeline developed in this project successfully automates the retrieval, transformation and storage of PubMed data, providing a scalable and efficient solution for analyzing medical research trends. By using PostgreSQL's snowflake schema design, the system enables fast and flexible querying, while Power BI facilitates user-friendly visualization and dynamic exploration of the processed data. However, the system's feasibility and limitations must be critically assessed to guide future enhancements.

System Feasibility

The pipeline efficiently handles the retrieval and transformation of thousands of articles, reducing manual effort. 30000 articles take about 5 minutes to fetch, depending on the user's internet connection bandwidth and the load on PubMed's servers. Query execution times in PostgreSQL remain fast due to the simplicity of the snowflake schema design, even with increasing data complexity. The system is built to scale, allowing new MeSH terms, medical fields, or additional metadata to be added easily. Optimizing the database structure and using indexing helps handle larger datasets efficiently.With dynamic filtering and multi-dimensional queries in Power BI, users can explore data and gain useful insights. Examples include analyzing how often MeSH terms appear and tracking trends in specific journals.

Challenges and Limitations

The API imposes rate limits of 10 requests per second (if API key is provided), which can slow large-scale data retrieval. While batching requests partially solves this issue, the API's limitations remain a bottleneck for processing larger datasets.

PubMed searches rely on MeSH term matching, which may result in irrelevant articles being retrieved. For instance, searches for "AI" and "cancer" may return articles unrelated to AI-driven cancer detection. This limits the precision of the dataset without additional refinement techniques.

Some articles lack complete metadata, such as missing or non-standardized country names, publication dates or authors. Basic data cleaning solved this issue to some extent, but certain metadata, such as country names, would require advanced preprocessing techniques like entity recognition to provide consistency.

While PubMed provides access to a vast collection of biomedical literature, it does not include all relevant sources. This reduces the comprehensiveness of the dataset, particularly for cutting-edge research in emerging fields like AI. Expanding data sources to include Scopus, IEEE Xplore, or Google Scholar would enhance the dataset's coverage.

Many articles lack author metadata. This creates complications when analyzing author contributions and identifying key researchers.

Potential Improvements

Some improvements can be implemented to assure more comprehensive and reliable results:

- Introduce natural language processing (NLP) techniques and large language models (LLMs) to refine article selection, ensuring higher relevance to chosen search terms. LLMs can also address complex metadata issues, such as normalizing country names (e.g., “USA” vs. “United States”) and filling missing information based on context.
- Incorporate additional bibliographic databases, such as Scopus, IEEE Xplore and Google Scholar, or use web scraping to include relevant articles from medical news websites. This would broaden the scope of analysis and improve the dataset’s representativeness.
- Enhance the pipeline to enable real-time or periodic updates, ensuring that the data remains current and supports ongoing research monitoring.

Conclusions

The results confirm that the system’s pipeline is able to successfully retrieve, processes and store PubMed data in a structured format, enabling efficient querying and trend analysis. The system allows automated data collection, reducing manual effort while maintaining consistency and accuracy in biomedical research metadata. The PostgreSQL database, structured using a snowflake schema, ensures efficient data management, allowing flexible filtering and analysis. The schema design supports scalability, enabling the integration of additional search terms, metadata fields and potential data sources without significant modifications. Power BI visualizations confirm that the stored data is accessible and usable for trend analysis. The interactive dashboards effectively illustrate MeSH term frequency trends, publication volumes across medical specialties and author contributions, demonstrating the system's value for biomedical research analytics. While the system functions as intended, there are areas for future enhancement. The inclusion of natural language processing (NLP) techniques could improve search precision by refining article selection. Additionally, large language models (LLMs) could assist in normalizing metadata such as country names. Expanding data sources beyond PubMed to include Scopus, IEEE Xplore and medical news websites would further improve the dataset.

Overall, the system is functional, scalable and provides a strong foundation for further development. Future improvements could refine filtering mechanisms, enhance metadata processing and optimize API retrieval for handling larger datasets, ensuring even greater reliability and insight generation.

Appendix

The complete code for this project is provided as separate files included with the digital submission of this thesis. The files are titled:

- "01_Data_Fetching_and_Cleaning.ipynb"
- "02_Database_Creation.ipynb"

Table of figures

References

[1] N. L. o. Medicine, "Nationa Library of Medicine," 30 04 2024. [Online]. Available: https://www.nlm.nih.gov/bsd/medline_pubmed_production_stats.html. [Accessed 31 01 2025].

[2] N. L. o. Medicine, "MEDLINE, PubMed, and PMC (PubMed Central): How are they different?," 28 12 2023. [Online]. Available: https://www.nlm.nih.gov/bsd/difference.html. [Accessed 31 01 2025].

[3] N. L. o. Medicine, "MEDLINE®/PubMed® XML Data Elements," 28 02 2018. [Online]. Available: https://www.nlm.nih.gov/bsd/licensee/data_elements_doc.html. [Accessed 31 01 2025].

[4] M. V. L. M. D. S. R. S. P. M. G. D. Santoro, *Web Application Programming Interfaces (APIs): general-purpose standards, terms and European Commission initiatives,* Via Enrico Fermi, 2749 — 21027 Ispra (VA), Italy: European Commission, Joint Research Centre (JRC), Digital Economy Unit (JRC.B6), 2019.

[5] N. L. o. Medicine, "A General Introduction to the E-utilities," 12 12 2022. [Online]. Available: https://www.ncbi.nlm.nih.gov/books/NBK25497/. [Accessed 31 01 2025].

[6] E. Sayers, "E-utilities Quick Start," PubMed, 24 10 2018. [Online]. Available: https://www.ncbi.nlm.nih.gov/books/NBK25500/. [Accessed 07 02 2025].

[7] Elsevier, "Elsevier Developer Portal," [Online]. Available: https://dev.elsevier.com/. [Accessed 01 02 2025].

[8] Springer, "Springer Nature Data Solutions," [Online]. Available: https://dev.springernature.com/#api. [Accessed 01 02 2025].

[9] GeeksforGeeks, "Database Schemas," 13 01 2025. [Online]. Available: https://www.geeksforgeeks.org/database-schemas/. [Accessed 01 02 2025].

[10] L. Siddiqui, "What is a Database Schema? A Guide on the Types and Uses," DataCamp Inc., 30 08 2024. [Online]. Available: https://www.datacamp.com/tutorial/database-schema. [Accessed 2025 02 01].

[11] M. M. Kirmani, "Dimensional Modeling using Star Schema for Data Creation," *Oriental Journal of Computer Science and Technology,* vol. 10, no. 4, pp. 745-754, 2017.

[12] P. Ponniah, Data Warehousing Fundamentals: A Comprehensive Guide for IT Professionals, New York / Chichester / Weinheim / Brisbane / Singapore / Toronto: John Wiley & Sons, Inc, 2001.

[13] J. D. U. J. W. Hector Garcia-Molina, DATABASE SYSTEMS The Complete Book (Second Edition), New Jersey: Pearson Education, 2009.

[14] E. V. G. A. L. M. Andrea Manconi, "Literature Retrieval and Mining in Bioinformatics: State of the Art and Challenges," *Advances in Bioinformatics,* vol. 2012, p. 10, 2012.

[15] N. G.-B. P. R.-M. A. T.-G. Miguel Pedrera-Jiménez, "TransformEHRs: a flexible methodology for building transparent ETL processes for EHR reuse," *ournal of Biomedical Informatics,* vol. 2022, pp. 89-102, 2022.

[16] M. G. K. B. M. K. T. Y. E. B. P. H. C. U. &. L. M. S. Toan C. Ong, "Dynamic-ETL: A Hybrid Approach for Health Data Extraction, Transformation, and Loading," *BMC Medical Informatics and Decision Making,* vol. 17, 2017.

[17] J. C. Luna, "Python vs R for Data Science: Which Should You Learn?," Data Camp Inc., 28 12 2022. [Online]. Available: https://www.datacamp.com/blog/python-vs-r-for-data-science-whats-the-difference. [Accessed 01 02 2025].

[18] DigitalOcean, LLC, "SQLite vs MySQL vs PostgreSQL: A Comparison Of Relational Database Management Systems," DigitalOcean, LLC, 11 03 2022. [Online]. Available: https://www.digitalocean.com/community/tutorials/sqlite-vs-mysql-vs-postgresql-a-comparison-of-relational-database-management-systems. [Accessed 01 02 2025].

[19] A. Biswal, "Power BI vs Tableau: Which Is Better Data Visualization Tool," 15 01 2025. [Online]. Available: https://www.simplilearn.com/tutorials/power-bi-tutorial/power-bi-vs-tableau. [Accessed 01 02 2025].