LLM-Based Researcher Search Engine

Dmitri Matetski
Hendrik Püss
Georg-Oliver Loorand

Repo link: https://github.com/TheGreyCore/LLM-based-researcher-search-engine/

Identifying business goals

Background

The idea of Meelis Kull gained support when it was informally discovered that there is a similar problem at the Clinic of the University of Tartu. Specifically, the medications have very numerous instructions (for example, how to dispose of them), and for a nurse to get an answer on how to do it correctly, she needs to search for a long time in the documents or ask colleagues. In our case, due to the lack of and inability to quickly obtain the necessary data, we settled on the proposed idea about researchers where we find researchers based on their descriptions.

Business goals

Create a LLM-based search engine which allows users to search for researchers by their research publications.

Business success criteria

The search engine allows a user to search for researchers by giving the program an input in natural language, and output researchers that fit the criteria.

Assessing the situation

Inventory of resources

As the main information provider we use ETIS. The university provides us with access to the OpenAI API.

Requirements, assumptions, and constraints

Risks and contingencies

Insufficient Data Availability

Cause: Limited availability of abstracts or incomplete datasets from ETIS.

Contingency Plan:

- Use web scraping to collect additional abstracts from open access sources like MDPI and other publishers.
- Collaborate with institutions to gain access to more comprehensive datasets.
- Integrate machine learning models to generate summaries for papers lacking abstracts.

Web Scraping Challenges

Cause: Websites with inconsistent formats, captchas, or anti-scraping measures.

Contingency Plan:

- Develop adaptive scraping scripts using libraries like selenium for dynamic content.
- Use proxy servers or rotating IPs to avoid getting blocked.
- Manually collect data from critical sources as a backup.

Quality Issues in Collected Data

Cause: Missing fields, encoding errors, or inconsistencies in scraped data.

Contingency Plan:

- Implement robust data cleaning pipelines using tools like pandas.
- Exclude incomplete entries during initial processing and focus on complete, high-quality records.

Delays in Team Coordination

Cause: Miscommunication, conflicting schedules, or insufficient task tracking.

Contingency Plan:

- Use project management tools like Trello for task assignments and updates.
- Schedule regular meetings for progress reviews.

Technical Failures

Cause: Bugs in software, crashes, or unexpected system issues.

Contingency Plan:

- Use version control systems like Git for efficient issue tracking and rollback.

Project Scope Creep

Cause: Adding new features or functionalities beyond the initial plan.

Contingency Plan:

- Define and document a clear project scope from the start.
- Prioritize core functionalities and defer non-critical features to future iterations.

Terminology

Search Engine - A software system designed to search for information on the internet or within a specific dataset and present relevant results to the user.

Web Scraping - The process of extracting data from websites using automated tools or scripts.

Vector Embedding - A numerical representation of text or other data, used to capture its meaning for computational tasks like similarity searches.

Abstract - A brief summary of a research paper or study, highlighting its objectives, methods, results, and conclusions.

Indexed Articles - Research articles listed in recognized databases, indicating a level of credibility and accessibility.

Data Availability - The extent to which required data is accessible and ready for use.

Classification - Grouping or labeling data into predefined categories based on specific criteria.

Selection Criteria - Defined rules and filters for including or excluding data in the dataset.

Filtering - The process of selecting data that meets specific conditions or criteria.

Dataset Completeness - The extent to which all required fields and information are present in a dataset.

Costs and benefits

Costs:

Licenses and Tools:

OpenAI API, estimated €100/year

Infrastructure costs, estimated €100/year

Benefits:

Time Saved in Finding Relevant Papers: If 1,000 researchers save an average of 10 hours per year by using the search engine, and their hourly rate is €20: €200 000/year.

Revenue from academic publishers promoting their papers, estimated benefit: €1000/year.

If 200 organizations (universities, companies) pay €100/year: benefit: €20 000/year.

Defining data-mining goals

Data-mining goals

Search Model

- Develop a search engine powered by vector embeddings to provide contextually relevant search results.

Data Cleaning

- Automate the cleaning, formatting, and deduplication of data.
- Python scripts ensure all input data is ready for embedding and search indexing.

Vectorized Dataset

- Transform text data into vector embeddings for similarity calculations.
- A file or database with numerical vectors corresponding to each research paper and its author.

Data-mining success criteria

Search Accuracy

- Ensure search results are contextually relevant to user queries.
- Mean Average Precision (MAP) of at least 85%.
- Quantitative, based on search engine test results against a labeled dataset.

Gathering data

Outline data requirements

To create a search engine, we firstly need to collect data on research papers and studies. Each entry in the dataset must include the year, authors, title, the abstract of the study, and a link to find more information on the studies where necessary. We will use most of this information to create vector embeddings. The abstract is the most crucial as that will allow us to create a more accurate search engine as opposed to only using the titles.

Verify data availability

The data is collected from the Estonian Research Information System (Eesti Teadusinfo Süsteem, ETIS). It lists thousands of publications and other research data.

We initially needed to solve the problem of most publications lacking the abstract within the ETIS directory. From over 14,000 thousand entries, only around 70 abstracts are available. For better results, more abstracts would be needed. To gather more data, we would need to create a webpage scraper. However, not all websites are built the same and it is not feasible to scrape all possible abstracts. We encountered multiple instances of faulty links to websites or missing links in general, and websites not being available due to content being moved or deleted.

To create the scraper, we are using bs4 from the Beautiful Soup library. It allows us to scrape abstracts relatively quickly, but must be adjusted to work with different websites. Using the scraper or studies linked in MDPI, a publisher of open access journals, we collected 541 more entries. This process will have to be repeated with other publishers and websites.

Define selection criteria

We define our criteria by publication classification, institution, language, and open access status.

For classification we included only indexed scholarly articles (1.1, 1.3), peer-reviewed articles (1.2), dissertations (2.3), published research project reports or studies (2.5), and specific research publications (3.3). We only included publications from the University of Tartu and Tartu University Hospital. Open access status is another requirement, as well as only including studies in English.

Describing data

For the moment, we have over 600 entries with all the required information, out of over 14,000. We will continue gathering data by scraping websites to increase the pool.

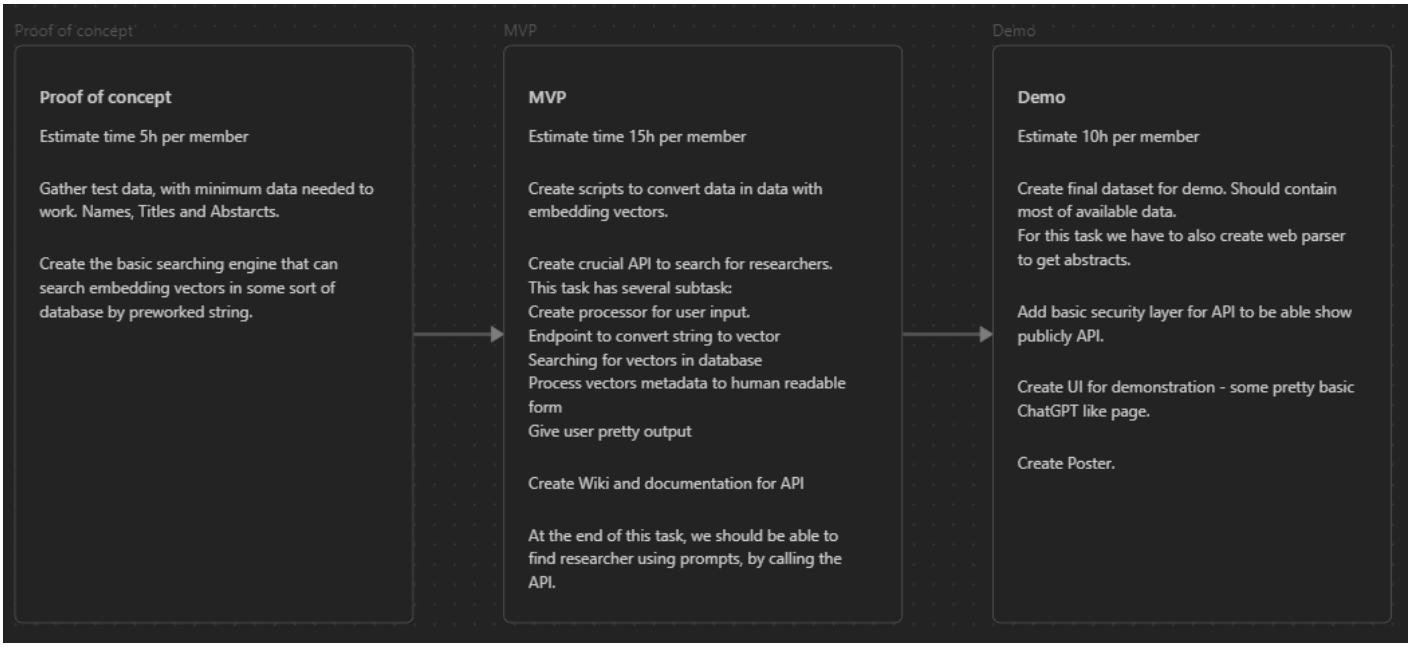
Exploring data

When collecting data, we performed an initial exploration into the general characteristics and possible inconsistencies of the dataset. As previously described, a complete dataset was not possible to export from ETIS. Instead, we are collecting more abstracts through scraping. Some entries have missing or incomplete information, and these will be removed.

Verifying data quality

The quality of our data is not perfect. Even with the language filter on English, some results come up in other languages. As well, a number of entries have formatting or encoding problems.

Task plan


Proof of concept
Proof of concept
Estimate time 5h per member
Gather test data, with minimum data needed to work. Names, Titles and Abstarcts.
Create the basic searching engine that can search embedding vectors in some sort of database by preworked string.
MVP
MVP
Estimate time 15h per member
Create scripts to convert data in data with embedding vectors.
Create crucial API to search for researchers.
This task has several subtask:
Create processor for user input.
Endpoint to convert string to vector
Searching for vectors in database
Process vectors metadata to human readable form
Give user pretty output
Create Wiki and documentation for API
At the end of this task, we should be able to find researcher using prompts, by calling the API.
Demo
Demo
Estimate time 10h per member
Create final dataset for demo. Should contain most of available data.
For this task we have to also create web parser to get abstracts.
Add basic security layer for API to be able show publicly API.
Create UI for demonstration - some pretty basic ChatGPT like page.
Create Poster.