Università della Svizzera italiana

Information Retrieval **2024**

Student: Costanza Rodiguez Gavazzi, Agnese Zamboni, Davide Frova

Due date: Wednesday, 27 November 2024, 11:59 PM

1. Overview

The Information Retrieval project is advancing according to plan. The responsibilities have been allocated among three team members, each concentrating on a separate part of the project.

2. Frontend Design and Implementation

Davide is responsible for the frontend design and implementation. The initial design research and UI mockup creation have been completed using Figma. The next steps involve implementing the design using Next.js. The design aims to be minimalistic and user-friendly. The current prototype contains simple search bar with the iconic search button, the length of the bar is not too short to incentivize the user to type longer queries. A logo representing the project will be placed above the search bar, this will aim to give the user a sense of the project's identity. Under the search bar we have the three pill shaped filter dropdowns, we identified as possible filters: the charity 'theme', the 'location' and the last filter is still to be decided. After entering the query the user will be presented with a list of cards cotaining various informations about the charities, such as the name, the logo and a "google snippet" like description with highlighted keywords. There are also present the 'thumb up' and 'thumb down' buttons to allow the user to give feedback on the results, this will re-run the query with the feedback in mind.
Here are some screenshots of the current design:

Figure 1: Mockup Search Page

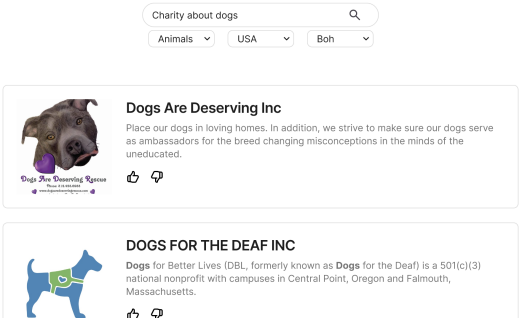

Figure 2: Mockup Results Browsing

3. Data Collection and Processing

1. Global Giving

Global Giving provides structured data on charitable organizations through its API in XML format. We utilized this API to collect comprehensive information about each organization, ensuring a reliable and standardized method of data retrieval.

1.1. Data Retrieval

To gather data, we accessed the Global Giving API and downloaded XML files containing details of various organizations. The use of their API eliminated the need for web scraping, allowing us to work directly with structured data. The XML files provided:

- Basic details such as the organization name and location.
- Operational metrics, including active and total projects.
- Mission statements and website URLs.

- Themes representing the causes they work on.
- Countries where the organizations operate.

1.2. Data Processing

After retrieving the data, we developed a parser using Python's `ElementTree` library to extract information from the XML format. To enhance the dataset:

- Geographical information was added using `pycountry` and `pycountry_convert`, allowing us to classify organizations by continent based on their headquarters.
- Field names were standardized to align with the data structure of Charity Navigator, ensuring consistency across platforms.
- The final dataset was converted to JSON format for easier storage and readability.

2. Charity Navigator

Charity Navigator offers data through its GraphQL API. Unlike Global Giving, Charity Navigator does not provide organization logos, requiring additional processing to locate this information.

2.1. Data Retrieval

Using the GraphQL API, we tailored queries to retrieve:

- Organization details, including names and locations.
- Ratings and operational metrics.
- Mission statements and website URLs.

The GraphQL API's flexibility allowed us to avoid redundant data requests and efficiently handle nested data structures. 10,000 records were retrieved in batches of 10 to not flood the server.

2.2. Data Processing

The lack of logos in Charity Navigator's data required us to develop a custom solution for locating and extracting organization logos from their websites. This system:

- Parsed webpage structures using `BeautifulSoup` to identify elements marked as logos.
- Checked metadata and special tags (e.g.,) for logo information.
- Scanned for images commonly used as logos, filtering out irrelevant elements like favicons or menu icons.

Standard Python libraries, including `requests`, `BeautifulSoup`, and `re`, facilitated these operations. Finally, the processed data was converted to JSON format to match the structure of Global Giving's dataset.

The Charity Navigator system enabled detailed data extraction while addressing the challenge of missing logos. The resulting dataset, enriched with logos and stored in JSON format, supports comprehensive analysis and cross-platform comparisons.

4. Backend and Indexing/Retrieval

Agnese is focusing on the backend and indexing/retrieval part. We have chosen PyTerrier for indexing and searching the data, and FastAPI to create a simple Python backend that will interface with PyTerrier. The necessary endpoints for the frontend-backend communication are being defined. An draft of the main endpoint comprehends parameters for the user query, the filters and the feedback on a result.

```
GET /search?q=userQuery&filters=filter1,filter2,filter3&feedback={docId:123,score:1}
```

5. User Evaluation

6. Appendix