Università della Svizzera italiana

Information Retrieval **2024**

Student: Costanza Rodiguez Gavazzi, Agnese Zamboni, Davide Frova

Due date: Friday, 20 December 2024, 11:59 PM

1. Overview

2. Frontend Design and Implementation

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

// todo add jupyter notebook etc 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

5. User Evaluation

6. Appendix