CONCEPTUAL DESIGN AND IMPLEMENTATION OF A PROTOTYPE SEARCH APPLICATION USING THE OPEN SEARCH INDEX

A. Nussbaumer[1*], R. Kaushik[1,2†], G. Hendriksen[3] S. Gürtl[1], C. Gütl[1]
[1]Graz University of Technology, Graz, Austria
[2]University of Waterloo, Ontario, Canada
[3]Radboud University, Nijmegen, Netherlands

Abstract

The development of special-purpose search engines requires crawling and indexing infrastructure, which needs specific technological knowledge and resources. This paper presents a the concept and implementation of a prototype search application that enables to create own search applications using the OpenWebSearch.eu index. The concept consists of the integration of an index partition exported from the Open Web Index and a search service that builds on Apache Lucene and offers a REST API to allow web applications to make use of it. A prototype implementation has been conducted that applies the conceptual design. This implementation can be used and modified if needed to create an own serach application. To demonstrate the concept and implementation, two example applications have been developed.

INTRODUCTION

In contrast to general-purpose search engines like Google, vertical search engines enable focused search in specific domains and allow domain-specific search operations. Current popular vertical search solutions are mostly commercially focused or integrated into enterprises' business models, such as Amazon's product search, LinkedIn's people search, or Booking.com's hotel search.

A vertical search engine (also called search application) needs an search index, which requires a lot of technological resources if newly created even for a fraction of the global web content. The OpenWebSearch.eu project aims to provide unbiased, democratic, and free search across the internet through its open access to its Open Web Index (OWI). In particular it allows to download a fraction or partition of the index, which can be used to to create a search application [1].

This paper describes the conceptual design of a vertical search engine and its integration with the OWI. Furthermore, it describes the implementation of a prototype search engine based on this concept. The paper seeks to demonstrate and provide a technological basis how a vertical search engine can be developed based on the OpenWebSearch.eu.

CONCEPTUAL DESIGN

The overall concept (see Figure 1) of a vertical search engine consists of two parts, the OWI and the search application. The term *search application* is used for the stand-alone search component with imported index partition.

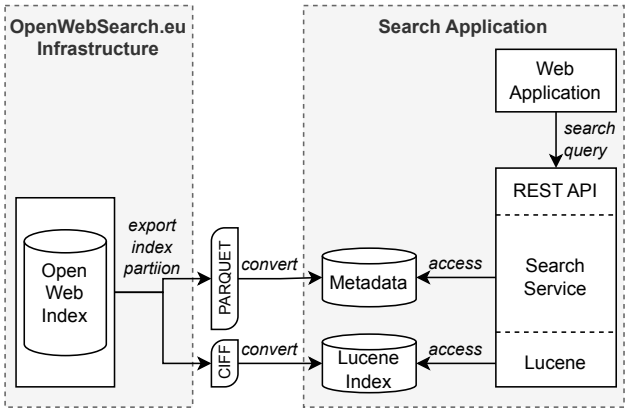

Figure 1: Conceptual design of a vertical search engine.

The OWI contains a vast corpus' of data collected from the World Wide Web and maintained it in data centers distributed throughout Europe. It allows to download partitions of the index that can be incorporated by a search application. An index partition is structured as Common Index File Format (CIFF) [2] and Apache Parquet[1]

CIFF aims to enable sharing and utilization of inverted indexes across different search systems. It provides a standardized format for representing index data, allowing multiple search engines to access and utilize the same index files. This format is quite useful in academia for searching indexes from various information retrieval systems and comparing their performance. However in an industrial setting, it cannot directly be used by most search libraries due to their own internal index formats. Therefore, in order to utilize the CIFF index, third party developers must convert it to the internal index format of the search library that they use. To resolve this issue, CIFF-Lucene converter[2] developed by Radbout university can be used to generate an index that can be used by Apache Lucene. Lucene has been chosen, since it is used as search engine library by commonly used search engine systems, sucah as Elasticsearch and Apache Solr.

In addition to index data, the Open Web Index also provides metadata of webpages in Parquet format. When creating the OpenWebSearch.eu index, the original content of the websites goes through common pre-processing steps like stop-word removal, stemming, lemmatization, lower casing

* alexander.nussbaumer@tugraz.at
† rohit.kaushik@uwaterloo.ca

[1] https://parquet.apache.org/
[2] https://github.com/informagi/lucene-ciff

and so on. However, snippets of original website data in conjuction with their links make for richer results in search applications. In order to preserve this original data, full text is stored in the metadata. Furthermore, the original metadata of the web pages and data stemming from the page analysis data are stored. These metadata are exported along with the index data in the Parquet format, which is a columnar storage file format widely used in big data processing and analytics frameworks such as Apache Hadoop and Apache Spark. It is designed to optimize I/O performance for large-scale data processing. The columnar data storage has several benefits including columnar compression, schema evolution without dataset rewrites and efficient queries by reading columns.

The core of the search service is the search application that coordinates the search and retrieval process. It provides a REST API that accepts search queries and returns search results. In order to perform the actual search, it makes use of Lucene that accesses the partitioned index converted to the native Lucene format. Using the metadata information, the prelimninary serach result can be limited and ranked, as well as enriched with fulltext snippets. The web application provides the user interface where search queries are created and results are displayed. This can be done in classic style with text field and link list, but also other forms are possible.

PROTOTYPE SEARCH APPLICATION

Based on the conceptual desgin of the OpenWebSearch.eu project, we created a prototype search application[3] that can be used by future search applications. This application includes a REST API that is intended for direct use by developers who wish to create search applications using the Open Web Index. The application requires developers to place their CIFF index along with its corresponding Parquet file into the `indexes` folder. It converts the CIFF index into an Apache Lucene index folder which forms the basis of the Search REST API. The API searches primarily over this Lucene index. The compressed Parquet file corresponding to the index is converted internally to a CSV file, using the Python library `parquet-tools`[4]. The CSV file is queried by the application through column pruning, allowing for efficient large scale data retrieval. This allows rich metadata to be returned by the REST API along with the index search results. The REST API is written in Java, and built using Apache Maven. The developer simply needs to run:

```
start_app.sh $index-name
```

This will first convert the CIFF index to a Lucene index and start the REST API at `localhost:8000`, and enable Cross Origin Resource Sharing (CORS) from `localhost:3000` by default. This allows any front-end application hosted on the latter to access and send requests to the API, independent of the API itself. The CORS and hosting ports can be changed from these defaults as required. The application also allows developers to retrieve an Apache Lucene index from the CIFF index by running a separate `convert_index` script. This allows developers to use this index with other search libraries of their choice, like ElasticSearch, Solr and Cassandra.

[3] https:// github.com/rkaushik29/java-lucene-search-api
[4] https://github.com/ktrueda/parquet-tools

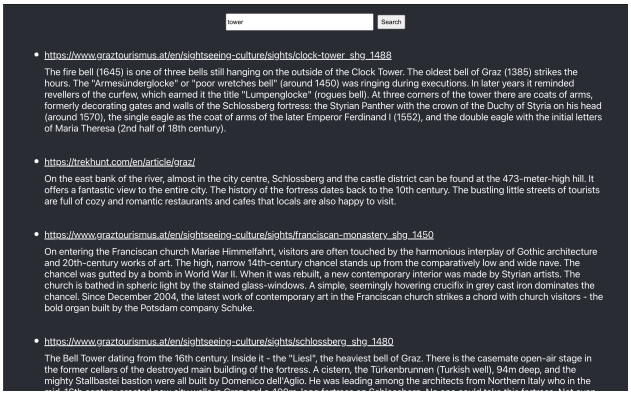

Figure 2: Graz Sightseeing Search App

DEMONSTRATION

The Prototype Search Application can be used to make any search application on the internet. To demonstrate this, we created a search application for popular sightseeing locations in Graz, Austria. The second demonstration is an application that allows to specify the search domain on which the search is executed.

Sightseeing Search

To create this application, a CIFF index file that indexed data of popular attractions in Graz was generated by the Open Web Search project. Using our index converter, the CIFF index was converted to an Apache Lucene index. The Parquet file containing URL metadata was also procured. These resources are sufficient and necessary to start the prototype application. Following this, the prototype application can be started and hosted on a server.

Simultaneously, a basic front-end application was created and hosted on a server using `React.js` which sends search queries as `GET` requests to the API. The API returns search results as URLs and metadata which are then displayed on the front-end. A visualization of the results is shown in Figure 2.

Topic-oriented Search

The nature of the prototype search application allows for the creation of topic-oriented search applications. If the application is given multiple CIFF index files along with their corresponding Parquet files, the user has the ability to choose the index that they want to perform the search in. This allows to chose the topic on which the search is performed by selecting the respective topic.

CONCLUSION

In a subsequent iteration, the Prototype Serach Application will be hosted directly by the Open Web Search Project and allow developers to access it by simply sending a query to a dataset. The process of generating a CIFF index and Parquet file, and searching over it can be done by this application internally, over one or more indexes. With the standardization of the Parquet file format across all Open Web Search data, the Universal API will become more robust and require no modification to serve users. An additional functionality of the API will be the possibility to search several indexes simultaneously - a useful feature for industrial applications. Finally, the API will be open source, to allow any developer to modify it and suit their requirements. This will be the first Universal Search API on the Internet, returning relevant links to websites and metadata in response to queries across multiple indexes. Further improvements to the API can be made upon understanding user requirements and feedback by conducting user studies.

ACKNOWLEDGEMENTS

This work has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101070014 (OpenWebSearch.EU, https://doi.org/10.3030/101070014).

REFERENCES

[1] M. Granitzer *et al.*, "Impact and development of an open web index for open web search," *Journal of the Association for Information Science and Technology*, in press. doi:10.1002/asi.24818

[2] J. Lin *et al.*, "Supporting interoperability between open-source search engines with the common index file format," *arXiv preprint arXiv:2003.08276*, 2020. doi:10.48550/arXiv.2003.08276