Data processing, indexing and querying

Web Information Extraction and Retrieval 2018/19, Faculty of Computer and Information Science, University of Ljubljana

Matej Klemen, Andraž Povše, Jaka Stavanja

***Abstract*—In this work we present our implementation of a simple index and queries againts it. First we are performing some data processing and indexing. With the gathered data and structured index, we are going to perform data retrieval by testing multiple queries and displaying the results. We test inverted index and naive approach and compare the performance of both.**

I. Introduction

Querying the world wide web is a task we perform every day. In the background, Google uses databases that contain indexed website data along with some other secrets to display most relevant results for the query at that specific moment. Therefore, in order to display relevant results, we must first process the information on the website, build an index based on it, and then use this information to display most relevant results. First step is processing the data and is described in chapter II. Next comes the indexing, which is presented in chapter III. Final step in the puzzle is data retrieval we perform after we are faced with a query. The implementation is shown in chapter V

II. Data processing

@Matej, opisi tist tvoj hack pri procesiranju oziroma indexiranju :D Separately describe data processing and indexing and data retrieval (with and without inverted index). Tokenization, stopword removal, ...

A. Inverted index

B. Naive approach

For the naive approach, we did not perform any a priori data processing, but handled it in real time. We tokenized each file, removed stopwords and then search for occurencies of the words that exist in the query.

III. Data indexing

Separately describe data processing and indexing and data retrieval (with and without inverted index). Indexing process, index occurency of each word, ..

A. Inverted index

B. Naive approach

No indexing was performed with naive approach. Each file was opened and processed when we ran a query.

IV. Database

Describe the database (number of indexed words, words and documents with the highest frequencies, . . .) **TODO: query the database for most frequent words, number of all words, etc.**

V. Data retrieval

TODO: Separately describe data processing and indexing and data retrieval (with and without inverted index). How we handled queries, searching words in our index, ...

Upon entering a query, we split it by ...

A. Inverted index

B. Naive approach

VI. Conclusion

We presented 2 different approaches to handle a query. First one used inverted index and was substantially faster than the other, naive implementation where we processed each file at time and had no built index. Building the index (SQLite database) did not take that much time compared to 0 with naive approach, but the end result when measuring time elapsed for each query was much better.