# Inverted index

Kacper Zaleski, Michał Dzienisik, Kamil Janowski

October 2022

## 1 Abstract

An inverted index is an index data structure storing a mapping from content, such as words or numbers, to its locations in a document or a set of documents. In simple words, it is a hashmap like data structure that directs you from a word to a document or a web page. It allows quick searching of text documents. The inverted index is a structure where, for each word, it is indicated the documents that contain the word. Thus, when a user enters a specific search term, it is very fast to know the documents that contain that term.

There are two types of inverted indexes: A record-level inverted index contains a list of references to documents for each word. A word-level inverted index additionally contains the positions of each word within a document. The letter form offers more functionality, but needs more processing power and space to be created.

Advantage of Inverted Index are:

- Inverted index is to allow fast full text searches, at a cost of increased processing when a document is added to the database.
- It is easy to develop.
- It is the most popular data structure used in document retrieval systems, used on a large scale for example in search engines.

## 2 Introduction

In order to have efficient search engine we need inverted index is to allow fast full text searches, at a cost of increased processing when a document is added to the database. This document consist of our implementation of word-level inverted index. It describes our solution to the problem and experiments done to test efficiency of script. Program is developed fully in C. Instructions how to use are in separate file. Program is deployed on github.

## 3 Solution

The solution is built for with an assumption for future modules as a crawler, searcher ex. In the main directory there is only Readme file and gitignore. going further in to the SearchEngine folder there is 'src' folder. Inside it there would be more services, but for now there is only 'SearchEngine.Calculation' that is a console application with purpose to index given files.

Indexer as a input value receive file and base information about it as: URL, title and data to index. Before indexing the data is analyzed by analyzer and filtered by character filter. Also there is a process of tokenization with is separating test in to words. And then tockenFilters are put in to data. Purpose of it is to remove all unnecessary data in the word. For example small letters, stop words or non letter or number characters.

# 4 Experiments

Testing are conducted on file with size 125kb and 20 000 words. And on the end of each one there is summary of total indexing time and number of indexes.

# 5 Conclusion

Index consisting with about 10000 characters translates to 600 word indexes and it gets calculated in 200ms, which is good result and more then enough to meet our search engine development goals.