Inverted Index Datamart Structure Experiments

Jorge Morales LLerandi
Denys Kavkalo Gumeniuk
Raquel Almeida Quesada
Gabriel Felipe Bernal Pinto
Adonai Ojeda Martín
Guillermo Cubas Granado

October, 20th

Abstract

This paper presents a comprehensive approach to building an inverted index for books downloaded from Gutenberg. The project includes a web crawler to download books, a cleaner to process the text, and an indexer to create both inverted and metadata indexes. The implemented solution uses two data structures for the inverted index to benchmark them.

1 Introduction

The rapid growth of digital libraries, such as Project Gutenberg, has increased the demand for efficient information retrieval systems capable of handling extensive textual data. This highlights the importance of developing advanced indexing techniques that ensure quick and effective querying of large datasets. In this context, the benchmarking of various programming languages becomes crucial for identifying the most efficient methods for text processing and indexing.

Numerous studies have explored different data structures for text indexing. This paper presents the implementation of an inverted indexing system for eBooks from Project Gutenberg, utilizing both trie and hashmap data structures. We benchmark the performance of these structures in Python, focusing on their efficiency in terms of speed and memory usage.

2 Problem Statement

Inverted indexing is a widely used technique in information retrieval systems, allowing for fast full-text searches. Traditional methods utilize hashmaps or lists, but more advanced structures like tries can enhance performance and scalability. This paper outlines the methodologies used in the implementation of the crawler, text cleaner, and indexer.

The challenge lies in processing large volumes of text data from Project Gutenberg, extracting meaningful information, and indexing it in a way that optimizes both storage and retrieval speeds.

3 Solution

The proposed solution includes several key components:

1. Crawler: A Python-based web crawler is designed to download books from Project Gutenberg, ensuring the data is in a manageable format.

2. Cleaner: The cleaner processes the raw text files, removing unnecessary stopwords, punctuation, and extracting relevant metadata such as title, author, and release date.

3. Indexer: The indexer constructs an inverted index using a trie data structure and a hashmap data structure. It also creates a separate metadata index for quick access to book information.

4. Query Engine: The query engine enables users to search for terms in the inverted index, returning the corresponding ebook numbers for each term.

4 Experiments

Experiments were conducted to compare the performance of different data structures for the inverted index. The trie was benchmarked against traditional hashmap implementations, focusing on query speed and memory usage. Results showed that the trie provided faster lookups for large datasets.

5 Conclusion

The project successfully demonstrates the effectiveness of inverted indexing for text retrieval from digital libraries. The implemented system is scalable and efficient, making it suitable for future enhancements.

6 Future Work

Future work will focus on optimizing the indexing algorithms and incorporating additional data structures to further evaluate their comparative performance. Additionally, a database system will be implemented to store the content of the JSON files, facilitating more efficient data retrieval and management. Exploring the implementation and testing of the system in other programming languages will also offer valuable insights. Moreover, plans include expanding the dataset and improving the crawler's functionality to handle a broader range of data formats and sources.

References

Project Gutenberg: `https://www.gutenberg.org/`

Python Documentation: `https://docs.python.org/3/`

JSON Documentation: `https://www.json.org/json-en.html`

Stack Overflow: `https://stackoverflow.com/`

ChatGPT: `https://chatgpt.com/`

GitHub Repository: Stage 1