Crawler Module and Inverted Index

Andrea Hernando González, Cynthia Quintana Reyes, Aarón Perdomo Aspas

November 2, 2022

Abstract

This article presents the development of the Crawler module. This module downloads a document every minute from Project Gutenberg and stores it in a document repository. This repository is chronological, it saves the documents by their download date in the yyyymmdd format. In addition, the project performs the inverted index of the documents downloaded in said repository. An inverted index is an index data structure that stores a protection from content, such as words or numbers, to their remainders in a document or set of documents. In simple words, it is a hashmap-like data structure that directs a word to a document or web page.
Our project is developed in Java, and is responsible for downloading a document from Project Gutenberg every minute and storing it in a repository. The Inverted Index is made of these documents, and returns us a dictionary that indicates the document and the lines of the same where each word is found.

Introduction

The programming language that we have used to develop this project has been Java, which is a cross-platform, object-oriented and network-centric language that can be used as a platform in itself. It is a fast, secure, and reliable programming language for coding everything from mobile applications and enterprise software to big data applications and server technologies.
To develop the code, we have used IntelliJ, which is an integrated development environment for software development. It is developed by JetBrains, and is available in two editions: Community Edition and Commercial Edition.
A crawler is a computer program used to automatically search and index website content and other information across the Internet. These programs, or bots, are most commonly used to create entries for a search engine index.
Most popular search engines have their own web crawlers that use a specific algorithm to collect information about web pages. Web crawling tools can be desktop-based or cloud-based. Some examples of web crawlers used for search engine indexing include the following: Amazonbot is the Amazon web crawler, Bingbot is Microsoft's search engine crawler for Bing...
The inverted index is a topic of relevance that has been studied for a long time by various scientists. Some interesting articles related to this topic are: "Inverted Index-Based Multi-Keyword Public Key Lookup Encryption with Strong Privacy Guarantee" by the authors: Bing Wang, Wei Song, Wenjing Lou, and Thomas Hou, and "Building an Inverted Index at the DBMS Layer for Fast Full Text Search" by the authors: Ciprian-Octavian Truica, Alexandru Boicea, Florin Radulescu.

This document explains how we have developed a document tracker and an inverted document index.

Methodology

The project is divided into different packages, on the one hand we have the API package, which contains the classes: APIException , which if the connection with the GutenbergApi does not work, throws an exception, the GutenberApi class, which creates a flyer to consume the api, the public static Metadata method makes a request to the api that returns a json with the book information, and then deserializes it.

Lastly, the Inverted Index class removes stopwords and special characters from the text and returns a dictionary with the location of each word.
In the Model package are the classes we need for the application model, The Document class that represents the document, the Document Builder and Metadata builder classes, the Metadata class that contains the MetaData attributes. The Resource class, which is an attribute of the MetaData and the SplitText class that separates the document into five parts (Metadata, beginning of the document, content, end of the document, bibliography). The Serialize package contains the Json class that deserializes the text.
The StoreDocs class stores documents in a directory in the format yyyymmdd.

Future work

For the next installment, we will take care of developing the search engine API. To develop this module, we will build a mock word-level inverted index, design API inputs and outputs (parameters and jsons, respectively) and deserialize the index from disk.
We will also modify the current delivery, so that on that occasion we collect the metadata in a more precise way. In addition, we will continue to improve the inverted index, with the intention of saving data and organizing it using Hashmaps, making its classification more accurate.

Conclusion

Crawlers are mainly used to collect data from other websites with which to create a database. To extract the data, the different search engines that analyze the sites and give them a position in the SERPs, among other things, are used.
The inverted index is a relevant technique as it allows you to store a mapping of content, such as words or numbers, to their locations in a document or set of documents. In simple words, it is a hash map-like data structure that directs you from a word to a document or a web page.
Our project is responsible for downloading a Project Gutenberg document every minute and storing it in a repository. The Inverted Index is made of these documents, and returns us a dictionary indicating the document and the lines of it where each word is found.
In conclusion, this program helps us to download documents accurately and to find the location of each word in said documents.