Finding Similar Book Reviews

Algorithms for Massive Datasets (2024/2025)

Antonios Tsipoulakos
Erasmus Student in Computer Science, University of Milan

June 24, 2025

Abstract

This project implements a basic similarity detector for Amazon book reviews. Using Jaccard similarity from scratch, we identify review pairs with high lexical overlap. The code is scalable and executable in Google Colab, supports dataset subsampling, and meets the guidelines of the "Algorithms for Massive Datasets" course.

Dataset Description

The dataset used is the **Amazon Books Reviews** dataset, available publicly under the CC0 license on Kaggle:
`https://www.kaggle.com/datasets/rajeevw/amazon-books-reviews`

The version accessed was downloaded in **June 2025**. We used only the file `Books_rating.csv`, specifically the column `review/text`.

Data Preprocessing

We applied the following preprocessing steps:

- Removed rows with missing values in the `review/text` field.
- Converted all reviews to lowercase and tokenized them into sets of words.
- Removed punctuation and trivial tokens (e.g., single characters).
- Introduced a global flag `USE_SUBSAMPLE` to enable execution on subsets (e.g., 100 or 500 reviews).

Algorithm Implementation

The Jaccard similarity was implemented **from scratch**, without external libraries. Given two sets of words A and B, the Jaccard similarity is computed as:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

All review pairs are compared in a brute-force manner. The top-5 most similar review pairs (based on this metric) are returned along with their scores and content snippets.

Scalability

To ensure scalability and reasonable execution time:

- A global flag controls dataset size (subsample vs full).
- The logic supports processing larger datasets without modification.
- Code is compatible with Colab, including automatic package installation and Kaggle dataset download.

Experiments

We ran the algorithm on a subset of **100 reviews**, computing all pairwise combinations ($\binom{100}{2} = 4950$ pairs).

The top 5 most similar pairs were printed with their scores and truncated text.

Results and Discussion

The method successfully identifies reviews with high word overlap. Despite its simplicity, the Jaccard index provides a good lexical similarity baseline. Limitations include:

- No handling of word semantics or synonyms.
- Tokenization may be too naïve for noisy text.

Possible extensions include:

- Using TF-IDF weighting
- Employing embedding-based similarity (e.g., sentence transformers)
- Parallelizing the brute-force loop

Academic Declaration

I declare that this material, which I now submit for assessment, is entirely my own work and has not been taken from the work of others, save and to the extent that such work has been cited and acknowledged within the text of my work. I understand that plagiarism, collusion, and copying are grave and serious offences in the university and accept the penalties that would be imposed should I engage in plagiarism, collusion or copying. This assignment, or any part of it, has not been previously submitted by me or any other person for assessment on this or any other course of study. No generative AI tool has been used to write the code or the report content.