Finding Similar Book Reviews using Jaccard Similarity

Hilal Mente - 47252A

Academic Year 2024/25
Master in Data Science for Economics

1 Introduction

This project's objective is to use scalable text similarity algorithms to find pairs of related book reviews. The dataset used for this purpose is the *Amazon Books Reviews* dataset, retrieved from Kaggle.

This project was completed by one person.In order to find reviews that share tokens and are lexically similar, I concentrated on Jaccard similarity method. My goal was to create an interpretable and computationally effective solution.

2 EDA

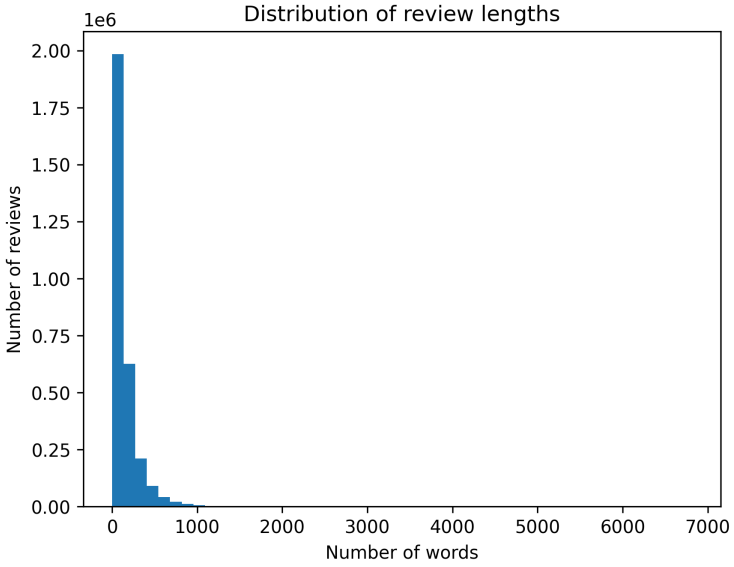

Figure 1: Histogram of review length distribution. This figure shows how many reviews fall into each word count range. The x-axis represents the number of words in a review, while the y-axis represents the number of reviews in that range. The majority of reviews are between 50 and 100 words, while extremely long reviews (greater than 500 words) are rare. This histogram helped guide our decision to limit the analysis to reviews with 200 words or fewer.

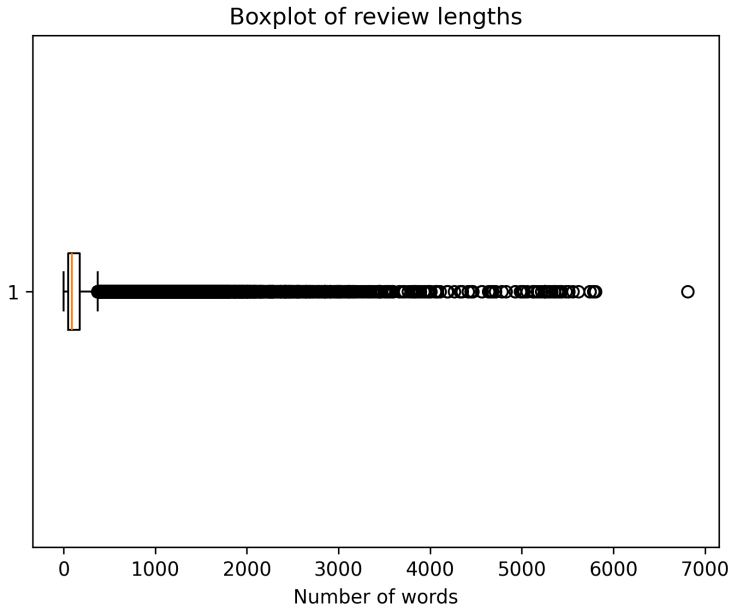

Figure 2: Boxplot showing review length distribution and outliers. The box represents the interquartile range (IQR), the line inside the box is the median, and the whiskers indicate the range of non-outlier values. Points beyond the whiskers are outliers—extremely long reviews, often over 1000 words. The median review length is around 50 words. This visualization reinforces that most reviews are short and justifies the removal of overly long reviews.

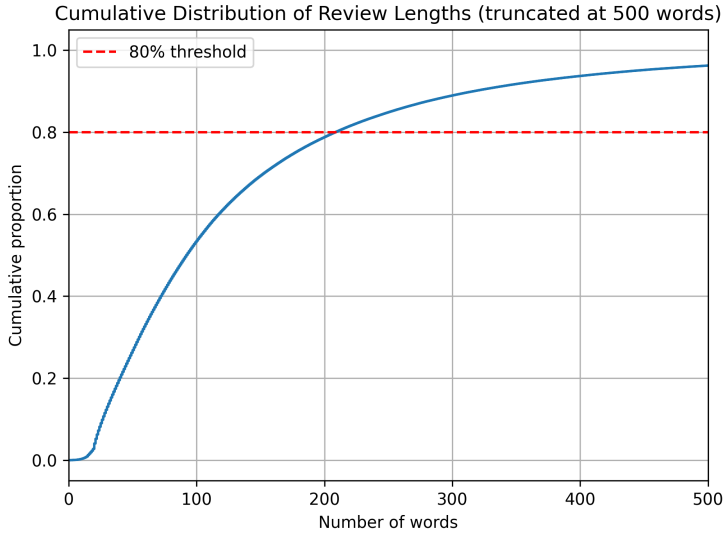

Figure 3: Cumulative distribution of review lengths (truncated at 500 words). This chart shows the cumulative proportion of reviews with lengths up to a given word count. The red dashed line indicates that 80% of the reviews are 200 words or fewer. This cumulative distribution informed the decision to use 200 words as a data-driven cutoff for filtering long reviews.

3 Dataset and Preprocessing

There are millions of reviews in the original dataset. I scaled up after choosing a random sample of 1000 reviews for experimentation. The following preprocessing steps were applied:

- Removing missing and empty reviews
- Removing punctuation and non-alphabetic characters
- Filtering reviews shorter than 200 words (for performance)
- Filtering reviews having count of word greater than 3. (for further analysis)
- Lowercasing all text
- Tokenizing on whitespace
- Removing exact duplicate reviews based on the cleaned text to avoid inflated similarity scores caused by identical entries.
- Removing HTML entities and normalized unicode artifacts to clean encoding-related noise from review text.

Timing and Performance

The preprocessing steps were timed to evaluate performance bottlenecks:

- **Download the CSV file:** $\sim$ 1 minutes
- **Full preprocessing (cleaning, filtering):** $\sim$ 1 minute 2 seconds
- **Full preprocessing (cleaning, filtering):** $\sim$ 1 minute 2 seconds

The most time-consuming of these was the entire preparation step, which included calculating word counts, cleaning text, and filtering long and short reviews. Given that the data was processed several times and regular expression operations were used, this is to be expected.

4 Jaccard Similarity Method

The ratio of two sets' intersection and union sizes is known as Jaccard similarity, and it is a set-based similarity metric formulated as:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{1}$$

where A and B are the sets of tokens (words) in two reviews.

I tokenized each review into a collection of lowercase words and then used Jaccard similarity on the cleaned text. This technique is helpful in situations when exact word overlap is significant since it catches lexical overlap. I limited the sample size throughout development in order to make the algorithm scalable.

5 Analysis and Discussion

Implementing and interpreting Jaccard similarity is simple. It does a good job of capturing lexical similarity at the surface level. It is limited, when reviews have similar meanings but differ in terminology. The decision of focusing on Jaccard was made for this project in order to preserve computational simplicity, interpretability, and transparency.

A boxplot and histogram were created to illustrate review lengths in order to better understand length-related skew. According to a cumulative distribution plot, 80% of the reviews are 200 words or less. This led to the selection of a new data-driven criterion that improves computing speed without compromising generalizability by excluding reviews that are too lengthy.

6 Experimental Results

I identified and analyzed the top review pairs based on Jaccard similarity. The validity of the method was confirmed by the fact that many of the pairs with the highest scores included significantly overlapping phrases and sentences. Initially, a few review pairs with a Jaccard score of 1.0 were observed due to duplicated content. These were filtered out to avoid biasing the similarity distribution and to ensure results reflected genuinely distinct but similar reviews.

Similarity Score Range and Interpretation

An analysis of the top 100 most similar review pairs revealed that the Jaccard similarity scores ranged between **0.5 and 0.24**. This relatively narrow range suggests that even the most similar reviews share only a moderate amount of lexical overlap.

This result highlights two key points:

- **Jaccard's strictness**: Since Jaccard similarity is based solely on word overlap, it can yield conservative scores, especially in natural language where word choice varies even among semantically similar texts.
- **Surface-level similarity**: Reviews in this dataset, while occasionally using common phrases, tend not to repeat exact wording extensively. Therefore, the similarity scores reflect partial overlap in sentiment-laden expressions (e.g., "highly recommend," "could not put it down") rather than full textual duplication.

In conclusion, a similarity score above 0.24 in this context still indicates a meaningful relationship between two reviews. However, more nuanced semantic similarity could be explored in future work using embedding-based approaches or transformer models.

Qualitative Analysis of Top 10 Similar Review Pairs

In this section, I present a qualitative inspection of selected highly similar review pairs detected using Jaccard similarity. Each pair is annotated with the Jaccard score and analyzed in terms of what contributed to the high similarity and whether the result was meaningful or potentially misleading.

Note: The full table is exported as a CSV file in the project repository including top 100 highest Jaccard similarity scores.

- **[288] vs [468]**
  Similarity: 0.50

Both reviews contain the phrase "recommend it to anyone". One is slightly more elaborate but otherwise carries almost the same message. This is a perfect example of Jaccard capturing short, high-sentiment overlap effectively.

- **[430] vs [977]**
  Similarity: 0.38
  Shared phrases like "book in good condition" and "timely manner" drive the match. However, the subject of recommendation shifts from "seller" to the "book" itself. Despite that, Jaccard picks up lexical overlap effectively.

- **[430] vs [839]**
  Similarity: 0.37
  Overlapping structure includes phrases like "good condition", "very pleased", and "timely manner". This again shows that Jaccard captures logistics-related satisfaction well, even if the core entities differ slightly (seller vs product).

- **[839] vs [844]**
  Similarity: 0.37
  A noisy pair. One review is long and fluent; the other is highly repetitive ("Great purchase" x5). Jaccard's inability to penalize excessive repetition means it considers this pair highly similar, though semantically it's weaker.

- **[839] vs [977]**
  Similarity: 0.36
  Both describe receiving the book in good condition, promptly. Lexical similarity is high and reflects meaningful similarity in user experience.

- **[150] vs [709]**
  Similarity: 0.32
  These reviews share the central theme "you will not want to put this book down". Although phrasing diverges slightly, they express identical enthusiasm for engagement. A solid semantic match detected by surface similarity.

- **[23] vs [519]**
  Similarity: 0.32
  Both contain phrases like "recommend this book", though their sentiment focus differs (delivery vs story). A mid-score that reflects some lexical overlap but limited contextual similarity.

- **[778] vs [851]**
  Similarity: 0.32
  Both reviewers declare the book as "one of the best I've ever read". This kind of superlative praise is common, and Jaccard effectively captures it.

- **[208] vs [778]**
  Similarity: 0.31
  Another "favorite book" pairing. Phrases like "best book" and "all-time favorite" align them lexically. Sentiment and general expression are quite parallel.

- **[328] vs [939]**
  Similarity: 0.31
  A meaningful semantic overlap: both are nonfiction, informative, and written in an accessible

way. Phrases like “easy to follow”, “informative”, “recommend” are shared. A well-deserved similarity match.

Overall, these examples show that Jaccard similarity is effective in finding stylistic and sentiment-based matches, but it may overvalue repetitive wording and cannot distinguish nuanced meaning changes. Some matches are strong and valid, while others hint at the limitations of pure lexical overlap.

7 Scalability

The implementation includes a global variable to toggle between full and sampled data for development and scalability assessment.Although the complexity of the current solution is quadratic, it can be enhanced in further work by employing other similarity algorithms.

8 Reproducibility

The notebook found at the associated GitHub repository can be used to replicate the project in its entirety. The code has comments, is readable, and is modular. A secured token is used to manage the data access through the Kaggle API.

To ensure experimental replicability, all preprocessing stages are established and managed via variables for cleaning, filtering, and sampling. The preprocessing pipeline also includes a deduplication step using the cleaned text to remove exact copies of reviews.

Also, to ensure consistency across runs, random sampling operations were controlled using a fixed random seed.

9 Conclusion

A quick and efficient method for identifying lexically similar reviews is Jaccard similarity. It accurately captures similarity for book reviews with overlapping wording or common terms. It also provides baseline performance and high transparency.

One potential limitation is that longer reviews may receive lower similarity scores due to the nature of the Jaccard metric. Future work could explore length-normalized or weighted similarity measures to address this.

Declaration of Originality

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, and including any code produced using generative AI systems. 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.