Review Similarity Analysis using Jaccard Distance

Ioannis Leivaditis

June 24, 2025

Abstract

This project performs a Jaccard similarity analysis on book reviews from the Amazon Books dataset. Using Natural Language Processing (NLP) techniques, we preprocess review texts, compute similarity scores between pairs, and extract the most similar review pairs based on shared vocabulary.

1 Introduction

Understanding the similarity between user reviews can improve recommendation systems and help detect redundancy or spam. This project uses the Jaccard similarity coefficient on tokenized reviews to identify closely related texts.

2 Dataset

The dataset used is the **Amazon Books Reviews** dataset, available on Kaggle at: `mohamedbakhet/amazon`
A subset of the reviews is used for analysis.

3 Methodology

3.1 Setup and Authentication

We authenticate with the Kaggle API to programmatically download the dataset:

```
os.environ['KAGGLE_USERNAME'] = "yannisleivaditis"
os.environ['KAGGLE_KEY'] = "MY_KAGGLE_KEY"
```

Listing 1: Kaggle API Authentication

3.2 Text Preprocessing

The text is converted to lowercase, punctuation is removed, and stopwords are filtered using NLTK:

```
def preprocess_text(text):
    text = text.lower()
    for p in string.punctuation:
        text = text.replace(p, '')
    return [word for word in text.split() if word not in STOP_WORDS]
```

Listing 2: Text Preprocessing Function

3.3 Jaccard Similarity

Jaccard similarity between two sets A and B is:

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

Implemented manually below:

```
def jaccard_similarity(tokens1, tokens2):
    intersection = []
    for w in tokens1:
        if w in tokens2 and w not in intersection:
            intersection.append(w)
    union = list(tokens1)
    for w in tokens2:
        if w not in union:
            union.append(w)
    if len(union) == 0:
        return 0.0
    return len(intersection) / len(union)
```

Listing 3: Manual Jaccard Similarity

3.4 Pairwise Comparison

We compute similarity for all unique review pairs and extract the top k:

```
for i, j in combinations(range(len(df)), 2):
    sim = jaccard_similarity(df.at[i, 'tokens'], df.at[j, 'tokens'])
    ...
```

Listing 4: Similarity Computation

4 Implementation Highlights

4.1 Dataset Download

```
os.environ['KAGGLE_USERNAME'] = "..."
os.environ['KAGGLE_KEY'] = "..."
os.system(f'kaggle datasets download -d {KAGGLE_DATASET} -p {
    ↪ DOWNLOAD_DIR} --unzip')
```

Listing 5: Kaggle API Setup

4.2 Validation and Tokenization

```
if not os.path.isfile(csv_file_path):
    raise FileNotFoundError(f"{csv_file_name} not found")

df = df[[REVIEW_COLUMN]].dropna().drop_duplicates()
df = df[df[REVIEW_COLUMN].str.strip() != '']
df = df.head(MAX_REVIEWS).reset_index(drop=True)
df['tokens'] = df[REVIEW_COLUMN].apply(preprocess_text)
```

Listing 6: Data Validation and Cleaning

4.3 Result Output

```
print(f"\n[Pair: Review {i} and Review {j}]")
print(f"Jaccard Similarity: {sim_score:.4f}")
```

Listing 7: Formatted Output Example

4.4 Modular Functions

The code is organized into modular components:

- `load_reviews()` – loads and preprocesses reviews.
- `jaccard_similarity()` – calculates similarity.
- `find_similar_reviews()` – compares pairs.
- `print_top_similar_pairs()` – outputs results.

Summary of Contributions

1. Automated download and preprocessing of review data.
2. Manual implementation of the Jaccard similarity metric.
3. Pairwise comparison and ranking of similar reviews.
4. Structured and readable output of top similar review pairs.

Future improvements may include optimizing performance using set operations, vectorization, or more advanced NLP libraries like SpaCy or scikit-learn.

5 Results

For a sample of 100 reviews, we identified the top 3 most similar review pairs based on Jaccard similarity. Below are the actual outputs from the analysis:

Pair: Review 24 and Review 29
Jaccard Similarity: 0.1690
→ Review 24: *I just finished reading Whisper of the Wicked saints. I fell in love with the caracters. I expected an average romance read, but instead I found one of my favorite books of all time. Just when I thought I could predict the outcome I was shocked! The writting was so descriptive that my heart broke w...*
→ Review 29: *I am an avid reader and I was shocked at how hooked I became on this book. I thought the first chapter was a little long and a little too discriptive, but truth be told after that I could not put this down. I read the other reviews on Whispers of the wicked saints before I wrote this and I saw one b...*

Pair: Review 52 and Review 68
Jaccard Similarity: 0.1429
→ Review 52: *This play was excellent. It's very smart, intellectually and morally meaty, and fast. I highly recommend it. Especially good material to ponder for people who in today's age can still think of the US or any country as being moral and right and good....*
→ Review 68: *A really good book to go with the software an excellent guide I highly recommend . . . .*

Pair: Review 24 and Review 31
Jaccard Similarity: 0.1404
→ Review 24: *I just finished reading Whisper of the Wicked saints. I fell in love with the caracters. I expected an average romance read, but instead I found one of my favorite books of all time. Just when I thought I could predict the outcome I was shocked! The writting was so descriptive that my heart broke w...*
→ Review 31: *I happen to love romance novels, but only if they are goos romance. I am not one who loves everything, but this made my heart rejoice. I absolutlely could not put it down. Wow what a book!! You have to read this awesome story of forbiden love. I warn you now. It is steamy...*

6 Code Repository

The source code for this project, is available on GitHub:

`https://github.com/yannisleiv1/algr`$_p$*roject/blob/main/main.ipynb*

7 Disclaimer

"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/our 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."