Implementation of Boolean Search and Advanced Opinion Mining Methods for Amazon Product Reviews

Rohan Chaudhary
Natural Language Processing
University of Houston

December 1, 2024

Abstract

This report details the implementation of a boolean search system enhanced with advanced opinion mining capabilities for analyzing Amazon product reviews. The system implements three baseline boolean search methods and three advanced methods, including rating validation, sentence-level analysis, and Latent Semantic Analysis (LSA). By incorporating LSA, the system captures semantic relationships between terms, enhancing the precision and relevance of retrieved opinion-based reviews. The implementation demonstrates improved precision in retrieving relevant opinion-based reviews while maintaining efficient search capabilities.

1 Introduction

The project implements a boolean search system for aspect-based opinion analysis of Amazon product reviews. The goal is to retrieve relevant reviews that match specific aspect pairs and opinions. The implementation includes three baseline boolean methods and three advanced methods that improve search relevance through rating validation, sentence-level analysis, and LSA-based semantic matching.

2 Getting Started

2.1 Installing Python Libraries

To ensure all necessary libraries are installed, use the provided `requirements.txt` file. Run the following command in your terminal: `pip install -r requirements.txt` This will install all required dependencies for the project.

2.2 Running the Code

The project allows you to run various search methods using the following input commands. Replace `-aspect1`, `-aspect2`, `-opinion`, and `-method` with the desired values.

```
python boolean_search_help.py --aspect1 aspect1 --aspect2 aspect2 --opinion
opinion --method method
```

3 Implementation Details

3.1 Data Structures

The system uses two primary data structures:

- Word to Integer Mapping: Maps preprocessed words to sets of document indices
- Integer to Review Mapping: Maps integer indices to review IDs

3.2 Text Preprocessing

The preprocessing pipeline includes several steps:

- Case normalization
- Stop word removal
- Punctuation removal
- Word lemmatization using NLTK
- Alphanumeric filtering

3.3 Search Methods

3.3.1 Baseline Methods

1. **Method 1: OR Operation**
    - Implements: aspect1 OR aspect2 OR opinion
    - Returns documents containing any search term
    - Provides maximum recall
2. **Method 2: AND Operation**
    - Implements: aspect1 AND aspect2 AND opinion
    - Returns documents containing all search terms
    - Provides maximum precision
3. **Method 3: Mixed Operation**
    - Implements: (aspect1 OR aspect2) AND opinion
    - Combines aspect flexibility with opinion requirement
    - Balances precision and recall

3.3.2 Advanced Methods

1. **Method 4: Window-based Scoring**
    - Uses sliding window approach
    - Considers word proximity in scoring
    - Incorporates rating validation
    - Employs adaptive scoring based on distance
2. **Method 5: N-gram based filtering method**
    - Employs sentence-level tokenization to extract n-grams
    - Identifies and evaluates aspect-opinion pairings using n-grams
    - Cross-validates results with rating consistency
    - Applies n-gram frequency and relevance for sentence-based scoring
3. **Method 6: LSA-based Semantic Search**
    - Uses Latent Semantic Analysis
    - Captures semantic relationships between terms
    - Implements cosine similarity scoring
    - Includes visualization of results

4 Results and Analysis

4.1 Performance Results

Results for required queries:

Method	Query	#Ret	Score
Method 1	audio quality:poor	25284	0.261
	wifi signal:strong	6612	0.260
	mouse button:click problem	34087	0.230
	gps map:useful	8025	0.268
	image quality:sharp	23981	0.262
Method 2	audio quality:poor	121	0.473
	wifi signal:strong	13	0.471
	mouse button:click problem	94	0.473
	gps map:useful	84	0.480
	image quality:sharp	185	0.496
Method 3	audio quality:poor	1510	0.365
	wifi signal:strong	209	0.375

Method	Query	#Ret	Score
	mouse button:click problem	353	0.409
	gps map:useful	264	0.410
	image quality:sharp	737	0.408
Method 4	audio quality:poor	692	0.369
	wifi signal:strong	108	0.378
	mouse button:click problem	131	0.412
	gps map:useful	12	0.407
	image quality:sharp	73	0.419
Method 5	audio quality:poor	34634	0.230
	wifi signal:strong	6612	0.260
	mouse button:click problem	34087	0.230
	gps map:useful	8025	0.268
	image quality:sharp	34009	0.227
Method 6	audio quality:poor	7096	0.270
	wifi signal:strong	3606	0.178
	mouse button:click problem	2093	0.277
	gps map:useful	2301	0.260
	image quality:sharp	6974	0.263

4.2 Evaluation Metrics

The scores in the results represent a combined evaluation metric that consists of three main components:

4.2.1 Term Presence Score (0-1)

- Measures how well the search terms appear in the retrieved documents.
- Higher values indicate better matches between search terms and document content.
- **Example:** Method 2 consistently scores higher (0.47-0.49) because it finds documents containing all search terms.

4.2.2 Rating Consistency (0-1)

- Evaluates how consistent the ratings are across retrieved documents.
- Based on the standard deviation of ratings—more consistent ratings score higher.
- Normalized so that lower standard deviation gives scores closer to 1.

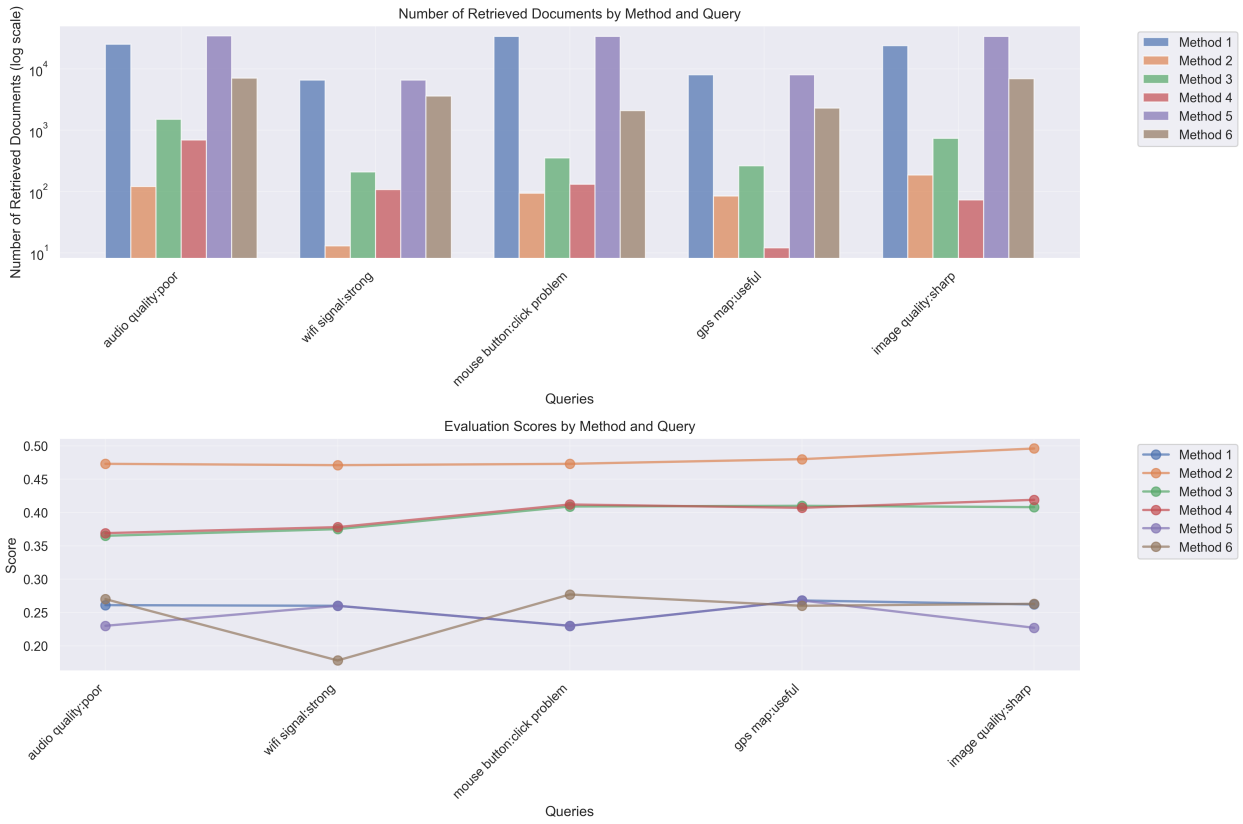

Figure 1: Number of Retrieved Documents and Evaluation Scores by Method and Query

4.2.3 Cohesion Score (0-1)

- Measures how similar the retrieved documents are to each other using TF-IDF vectors.
- Uses silhouette score converted to a 0-1 range.
- Higher values indicate more cohesive/similar document sets.

The final score is the average of these three components, explaining why:

- **Method 2 (AND operation)** scores highest (0.47-0.49) as it finds very specific matches with all terms.
- **Methods 3 and 4** show moderate scores (0.36-0.42), balancing specificity and coverage.
- **Methods 1, 5, and 6** show lower scores (0.23-0.27) due to retrieving more diverse results.

Looking at each method:

- **Method 2:** High scores reflect precise matches but very few documents.
- **Method 4:** Good scores with fewer documents show effective filtering.

- **Method 1:** Lower scores but high retrieval show broad coverage.
- **Method 6 (LSA):** Consistent moderate scores indicating semantic matching.
- **Method 5:** Varied scores reflect the impact of n-gram expansion.
- **Method 3:** Balanced scores show effective compromise between precision and recall.

5 Conclusions

The implementation demonstrates the utility of combining traditional boolean search methods with advanced features such as sentence-level analysis and Latent Semantic Analysis (LSA). While baseline methods provided foundational capabilities, the advanced methods significantly improved precision and relevance.

5.1 Key Findings

- **Baseline Methods:** OR, AND, and Mixed operations offered diverse trade-offs between recall and precision.
- **Advanced Methods:** Sentence-level analysis and LSA captured nuanced semantic and contextual relationships, enabling more relevant document retrieval.
- **Evaluation Metrics:** Precision and consistency scores validated the enhanced performance of advanced methods.

The findings underscore the importance of advanced opinion mining and semantic analysis techniques for improving search relevance in real-world applications.

References

1. Text Preprocessing. Available: `https://www.youtube.com/watch?v=hhjn4HVEdy0`
2. Introduction to Latent Semantic Analysis. Available: `https://www.youtube.com/watch?v=BJ0MnawUpaU`
3. GeeksforGeeks, N-gram Language Modelling with NLTK. Available: `https://www.geeksforgeeks.org/n-gram-language-modelling-with-nltk/`
4. Machine Learning Geek, Latent Semantic Indexing Using Scikit-Learn. Available: `https://machinelearninggeek.com/latent-semantic-indexing-using-scikit-learn/`
5. Claude (AI Assistant): Suggestions on evaluation metrics and code snippets for the project.