Building a Natural Language Opinion Search Engine

Roman Guerra
Department of Computer Science
University of Houston

I. Introduction

The growing volume of online reviews has transformed they way consumers and businesses interact. They provide insights into customer preferences, product quality, and overall experience. However, because of the abundance of data identifying relevant reviews for specific needs remains challenging. Consumers seeking opinions on a specific aspect/feature may struggle to sift through reviews, many of which may not be a relevant opinion.

This project focuses on building a Natural Language Opinion Search Engine capable of retrieving reviews based on aspect-opinion queries. By leveraging advanced Natural Language Processing (NLP) techniques, including Boolean Search, Sentiment Analysis, and Topic Modeling through Latent Dirichlet Allocation (LDA), this search engine addresses the precision in opinion retrieval.

This search engine is designed to understand queries combining aspects (e.g., "audio quality") with opinions (e.g., "poor") and filter results for relevance and sentiment.

A. Objectives

- Implement baseline and advanced search methods for retrieving relevant reviews.
- Evaluate methods using precision metrics and compare against a baseline.
- Analyze topic modeling's role in improving query relevance.

II. Dataset

A. Dataset Description

The dataset consists of 210, 761 Amazon product reviews related to electronic and software products, spanning 112,953 unique products reviewed by 50,704 customers. Each review includes metadata, such as star ratings, review titles, and text. Table I summarizes the dataset attributes.
The dataset occupies 24.1 MB in memory, and exhibits a diverse range of review lengths, sentiments, writing styles, and a comprehensive view of customer feedback enabling detailed analysis of textual reviews.

B. Exploratory Data Analysis (EDA)

To understand the dataset, various exploratory analyses techniques were applied on both the metadata and the textual content.

Textual Characteristics: The review_text field contains 210,761 reviews with a wide ranges of lengths. The median review length is 85 words, with the longest review containing 4553 words, and shortest review containing 1 word. Reviews included a diverse vocabulary of approximately 157,361 unique terms after processing. Figures 1 shows the distribution of review lengths, highlighting the dominance of long reviews.

Rating Distribution: The star ratings customer_review_rating provide a metric for sentiment and opinion analysis. The ratings are skewed toward higher values, with 888% of the reviews having 4 or 5 stars. Figure 2 visualizes this distribution.

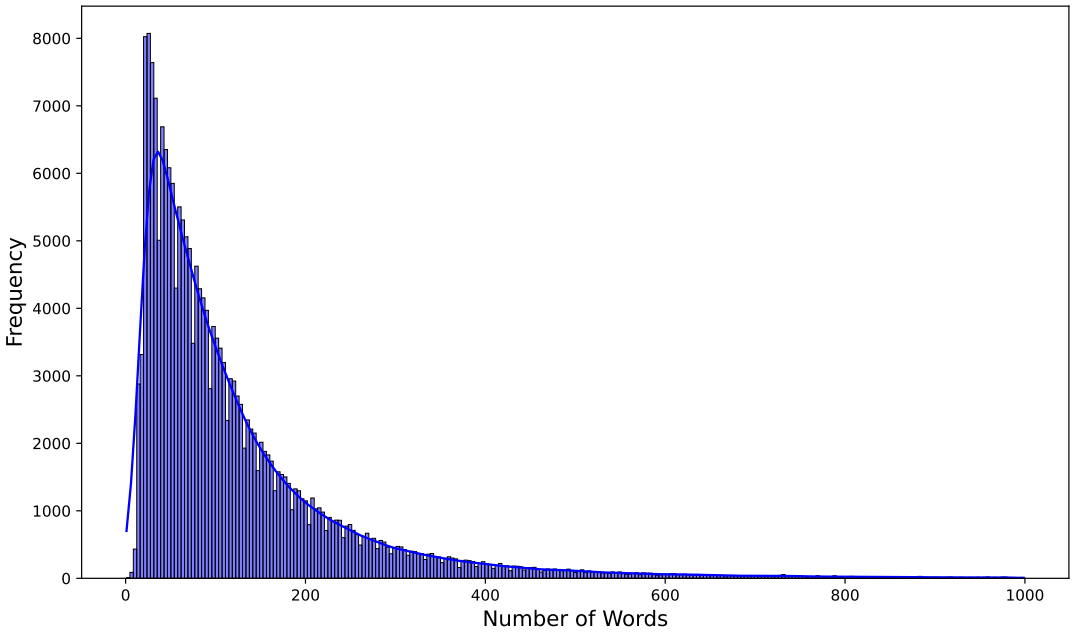

Fig. 1. Distribution of Review Word Lengths

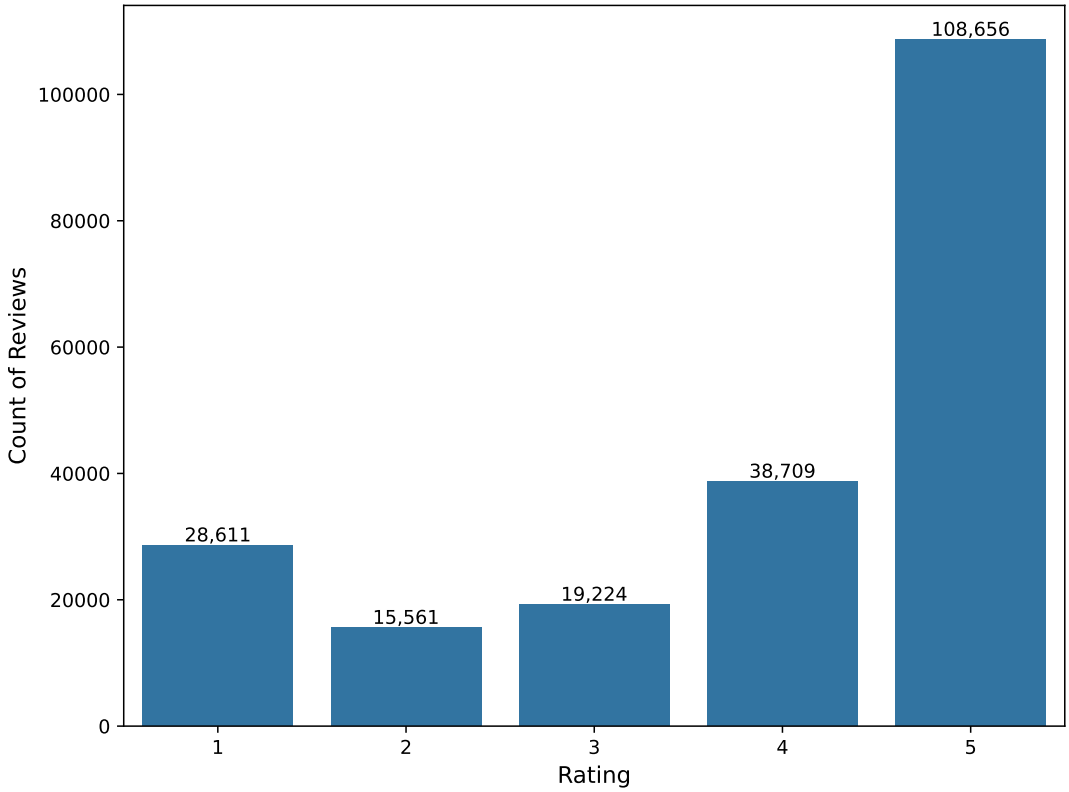

Fig. 2. Distribution of Review Ratings

TABLE I
DATASET ATTRIBUTES

Attributes	Description
review_id	Unique review identifier
product_id	Unique product identifier
customer_id	Unique customer identifier
review_title	Title of the review
review_written_date	Date of the review
customer_name	Name of the customer
review_text	Full review text
helpful_count	Number of helpful votes
out_of_helpful_count	Total votes for helpfulness
customer_review_rating	Rating given by the customer
number_of_comments	Total comments on the review
amazon_verified_purchase	Indicates if the purchase was verified
amazon_vine_program_review	Indicates if part of Vine program
review_with_metadata	Combined text and metadata

Stop Words and Lemmatization: Analysis of the text revealed that 44.06% of the total tokens are stop words, which were removed during preprocessing. Additionally, lemmatization was applied to 42.94% reducing the vocabulary size by normalizing variations like 'running' and 'ran' to their root form 'run'. Figure 3 plots their percentages.

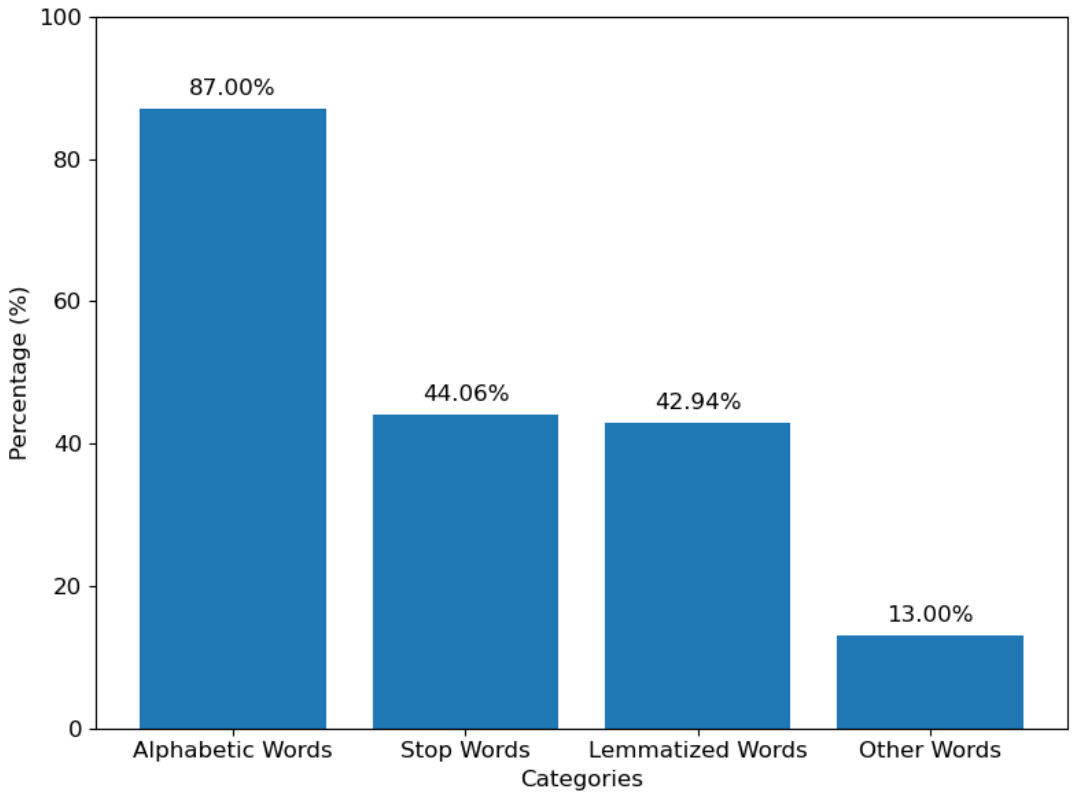

Fig. 3. Distribution of Review Ratings

Word Frequency: The most frequently occurring words provide insights into the dataset's primary focus. Figures 4 shows the top 10 most common words after preprocessing.

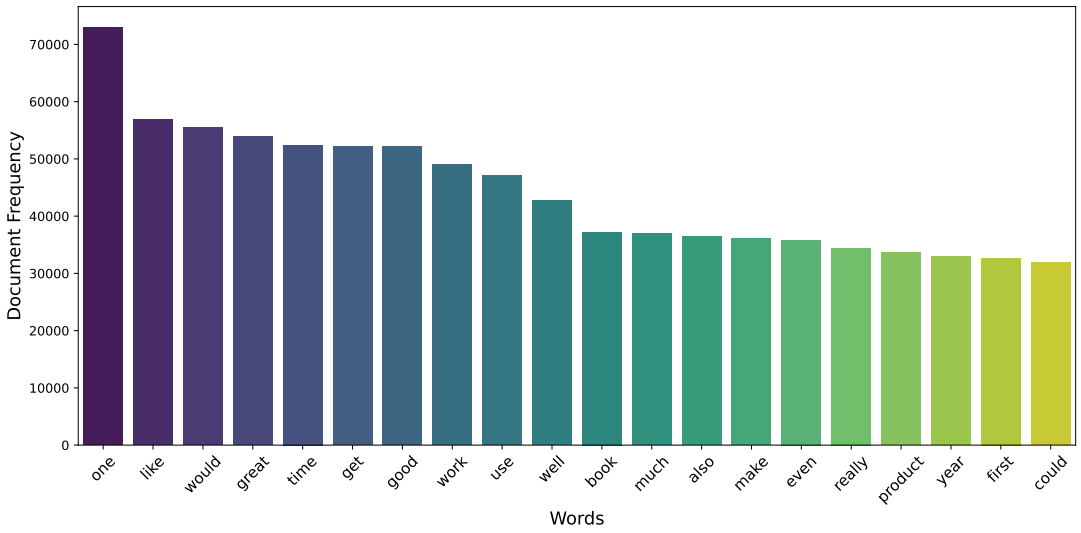

Fig. 4. 20 Most Common Words

C. Preprocessing Steps

The following preprocessing steps were implemented to prepare the data for analysis and focus on meaningful content:

1) **Punctuation Removal**: Removed non-informative symbols while retaining sentence-ending punctuation for sentiment detection.
2) **Link Removal**: Stripped hyperlinks and other extraneous text.
3) **Tokenization**: Split the review text into individual tokens for easier processing.
4) **Lowercasing**: Converted text to lowercase to ensure case-insensitive analysis.
5) **Lemmatization**: Standardized words to their base forms, reducing redundancy in the vocabulary.
6) **Stop Word Removal**: Eliminated uninformative words (e.g., ”the,” ”and”), which carry minimal semantic weight.
7) **Spelling Check**:

III. Problem Definition

The main problem addressed in this project is the efficient retrieval of relevant queries. The volume of online reviews make it difficult to identify relevant reviews that align closely with a customer's needs. This challenge is more difficult with the presence of noise in the data, irrelevant text, contradictory sentiments, and biased reviews. The goal is to create a search engine that can interpret user queries and return relevant reviews by considering semantic meaning, and sentiment.

A. Query Structure

Each query is structured using two critical components:

1) **Aspect:** A specific feature or characteristic of the product or service (e.g., ”battery life”, ”audio quality”).
2) **Opinion:** A subjective sentiment or evaluation of the aspect (e.g., ”poor”, ”excellent”).

For example:

- `(battery life: excellent)` Retrieves reviews where users discuss the battery life and describe it positively.
- `(audio quality: poor)` Retrieves reviews where users discuss the audio quality and describe it negatively.

B. Evaluation Metrics

The effectiveness of retrieval methods is evaluated based on:

1) **Total Documents Retrieved (Ret.)**: The total number of reviews retrieved by the search engine in response to a query.
2) **Relevant Documents Retrieved (REl.)**: The number of documents among the retrieved set that are relevant to the query. Relevance is determined bases on a combination of sentiment alignment, and manual validation.
3) **Precision:** Precision is calculated as:

$$\text{Precision} = \frac{\text{Rel.}}{\text{Ret.}} \tag{1}$$

It measures the proportion of retrieved documents that are relevant, emphasizing the accuracy of the search engine. High precision indicates that the search engine minimizes irrelevant results.

IV. Methodology

In this project, four methods were implemented to retrieve and rank reviews based on aspect-opinion queries. Each method incorporates different NLP techniques to enhance precision and relevance. Additionally, each method was tested using three query structures to explore the effect of query formulation on retrieval performance.

A. ***Method 1*** - *Baseline (Boolean Search)*

The first method, Boolean Search, establishes the baseline by leveraging an inverted index. Queries are matched exactly to the review text without considering sentiment or context.Search retrieves reviews containing specific keywords corresponding to aspects and opinions. This method provides a benchmark to evaluating more advanced methods.

B. ***Method 2*** - *Rating Search*

The second method augments Boolean search by integrating the review rating as an additional filter. Positive Opinion (e.g., ”excellent” are matched to reviews with a star rating greater than 3. Negative Opinion (e.g., ”poor”) are matched to reviews with a star rating of 3 or less. This approach uses a simple heuristic to align the query sentiment with rating distribution to improve relevance of the retrieved results.

C. ***Method 3*** - *Semantic Analysis*

In this method, semantic analysis is introduced using the TextBlob library. Sentiment polarity is calculated for each review using a Naive Bayes Classifier. The polarity score is in the range of [-1.0, 1.0]. Reviews sentiment polarity above 0 are classified as positive, while those below 0 are classified as negative. Queries are processed using Boolean Search, and retrieved reviews are filtered based on their sentiment classification. This method adds semantic understanding, but it depends on the accuracy of TextBlob sentiment analysis, which may vary across different context.

D. ***Method 4*** - *Topic Modeling Using Latent Dirichlet Allocation (LDA)*

The fourth method employs Latent Dirichlet Allocation (LDA) to identify topics in the review text. Reviews are assigned to specific topics bases on their content distribution. Queries are matched to these topics using their corresponding aspect-opinion keywords. Reviews belonging to the matched topics are retrieved, levering LDA's ability to group semantically similar content. This method provides an advanced concept for handling queries and uncovering patterns in textual data. Figures 5 shows topic proportions by topic name.

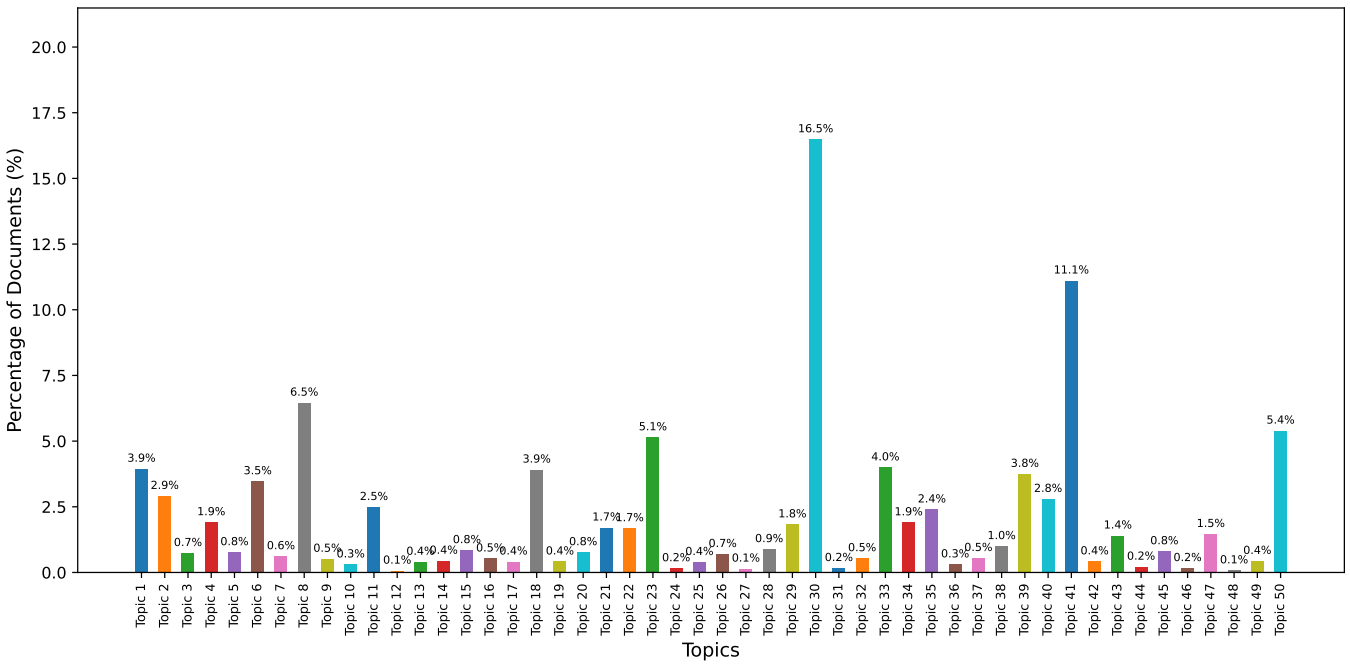

Fig. 5. 20 Most Common Words

E. Query Operation

Each method was tested using AND operation toe evaluate query formulation. AND operation on all terms retrieves documents that contain all the terms (aspect 1, aspect 2, opinion).

- Example: `audio AND quality AND poor`.

V. Implementation

A. Inverted Index

The inverted index is a core component of the search engine, mapping terms to reviews where they appear. This structure allows for efficient query evaluation, by quickly locating relevant reviews bases on the keywords in a query. The index was built by tokenizing and preprocessing the review text to clean text. Then sorting each term in a postings list. Finally, saving the inverted index for quick access during search.

B. Test Queries

- $q_1 = audio\ quality : poor$
- $q_2 = wifi\ signal : strong$
- $q_3 = gps\ map : useful$
- $q_4 = image\ quality : sharp$
- $q_5 = mouse\ button : problem$

C. Sample Query Text

To show the performance of the search engine, $q_1 = audio\ quality : poor$ was evaluated. Below are sample text from the reviews for q_1:

- **Baseline (Boolean Search)**: ”The major problem of this product is the audio quality. It’s simply poor when used within the local network, and totally useless if you try to use it over the Internet

even with high speed cable connection. If I knew that audio quality is that bad, I would buy IP100 and save $50."

- **Rating Search**: "In addition to the poor audio quality, I hadn't remembered the Weavers sounding so syrupy."
- **Semantic Analysis**: "I've had this for two weeks now. It has extremely poor audio quality and a constant loud buzzing noise that is even louder when used with the bluetooth headset. Returning it today because it's not usable for a conversation."
- **Topic Modeling**: "Awesome Quality for the Buck. Cancels noise very well, bass is top notch, and the durability of the ear bud so far has been good."

D. Results

Table II summarizes the retrieval performance for each query across four methods: Baseline (Boolean Search), Rating Search, Semantic Search, and Topic Modeling.

TABLE II
EVALUATION RESULTS FOR VARIOUS QUERIES ACROSS DIFFERENT METHODS.

Queries	Baseline (Boolean)			Method 1 (M1)			Method 2 (M2)			Method 3 (M3)		
	# Ret.	# Rel.	Prec.	# Ret.	# Rel.	Prec.	# Ret.	# Rel.	Prec	# Ret.	# Rel.	Prec.
audio quality:poor	128	61	0.48	128	11	0.09	128	64	0.50	128	15	0.12
wifi signal:strong	13	9	0.69	13	0	0.00	13	10	0.77	13	5	0.38
gps map:useful	91	36	0.40	91	19	0.21	91	82	0.90	91	39	0.43
image quality:sharp	189	6	0.03	189	180	0.95	189	186	0.98	189	121	0.64
mouse button:problem	245	179	0.73	245	136	0.56	245	227	0.93	245	174	0.71

The test results show that:

- **Baseline (Boolean Search)**: The baseline method struggled with precision as it retrieved reviews, matching keywords without considering context, or sentiment.
- **Rating Search**: Was effective for queries some queries like `wifi quality:sharp`, but not for others.
- **Semantic Analysis**: Outperformed other methods in capturing sentiment, making it precise for opinion queries.
- **Topic Modeling**: It effectively grouped related reviews for some queries, it did not perform well for other queries.

REFERENCES

REFERENCES

[1] Mukherjee, A. "Natural Language Processing Course Content." University of Houston, COSC 4397, Fall 2024.
[2] OpenAI. "ChatGPT" Accessed 2024. Available: https://openai.com/
[3] Bird, S., Klein, E., & Loper, E. "Natural Language Toolkit (NLTK): Python Libraries for Working with Human Language Data." Available: https://www.nltk.org/