Amazon Review Opinion Search Engine

In this new age of technology, customer reviews are a good source of information. However, these reviews often contain a mix of opinions about various aspects of a product, making it difficult to retrieve feedback on a specific feature such as wifi or audio quality. This project presents a system for retrieving aspect specific opinions from Amazon product reviews. We evaluate three different methods of retrieval: a Baseline keyword matching model, an advanced m1 model with rating based filtering, and an advanced m2 model that incorporates aspect-opinion proximity and sentiment polarity.

The main goal of this project was to implement and compare these three models to determine which is most effective at retrieving relevant opinions for five predefined queries. The project involved a combination of rule based filtering, text preprocessing, relevance evaluation, and precision calculation based on manual annotation.

Overview of Dataset

The dataset used consists of segmented Amazon product reviews, each with a unique review_id, review_text, review_title, and customer_review_rating. All text data was lowercase, and string formatting was applied to clean quotation marks and ensure consistency. Ratings were converted to numerical values to assist with filtering based on polarity.

Overall, thousands of reviews were available, though only the top 60 retrieved reviews per query per model were evaluated for relevance. This subset was sufficient for estimating model precision with low bias while avoiding evaluation fatigue.

Baseline Model: Boolean Term Matching

The Baseline model represents the most fundamental approach to information retrieval. It serves as a control method to evaluate how much more sophisticated techniques improve precision over simple keyword based methods. This model uses Boolean logic, specifically an “AND” query between aspect terms, to determine if a review is relevant to a given query.

Each query in our project consists of two main aspect terms, like audio and quality. For the baseline model what we did was make all the review texts lowercase for consistency. Then we checked if both aspect keywords are present anywhere in the text, and check if reviews that match both terms are included in the output file for that query.

This method is extremely lightweight and computationally cheap. It does not consider the position of words, sentiment, or any contextual meaning. It only checks for the co-occurrence of the keywords.

The strengths for this model is that it is simplistic and fast. Due to its simplicity, its logic can be easily improved or easy to debug and it has no dependency on any libraries. The limitations for this model, however, is that it is prone to false positives, meaning a review mentioning audio and quality in unrelated context is still counted. It has no sensitivity to polarity, positive or negative comments. There is no proximity check which can lead to low relevance in many cases.

The Baseline model generally showed lower precision in our evaluation, particularly for subtle or sentiment heavy queries like mouse button or wifi signal. However, in cases with broad keyword overlap, such as image quality, it occasionally performed surprisingly well.

Advanced M1: Full Text Keyword Co-occurrence

The Advanced M1 model expands on the Baseline model by introducing a more flexible matching mechanism. While still keyboard based, M1 makes improvements in how and where it searches for relevant content.

For each query, the same aspect terms such as mouse and button are used. Instead of a strict boolean AND, we check if the terms appear anywhere in either the review title or review body. We do not require that they appear in close proximity, which helps in some long form reviews but also leads to higher noise. We also consider common variations and plural forms by preprocessing the text more carefully, ensuring cleaner matches.

The strengths for this model is that it searches across both title and body, giving it a broader context. It is less rigid than the Baseline model and is still efficient for large scale use. The limitations for this model is that it has no contextual understanding of how terms are used. There is no sentiment filtering, which means it can misclassify neutral or irrelevant mentions as relevant. It also suffers when keywords are used in non-standard ways.

M1 improved precision in some domains but fell short in queries where sentiment or polarity mattered. For example, it did poorly in the wifi signal query, where many matches simply mentioned wifi without any evaluation of signal strength.

Advanced M2: Proximity and Polarity Aware Matching

The Advanced M2 model is our most sophisticated implementation and draws inspiration from aspect based sentiment analysis techniques in NLP. It combines keyword co-occurrence, proximity heuristics, and sentiment polarity filters to more accurately detect relevance.

Each query in M2 is defined with two aspect terms, and a set of opinion words such as useful or accurate, and a polarity label, positive or negative. A review is considered relevant if one or both aspect keywords are found in the title or text, an opinion word appears within a small word window of the aspect terms, and the sentiment polarity of the review rating matches the query. For example, positive polarity requires a review rating greater than or equal to 4 and a negative polarity accepts review rating less than or equal to 2. Then all matching is case insensitive and based on simple word splitting.

For example, for the query “wifi, signal, strong, positive”, we check if wifi and signal are found close together in the review and the word strong appears within a nearby window and the rating is greater than or equal to 4 stars. This approach mimics how humans interpret reviews. They do not just match terms but evaluate the context and tone.

The strengths of this model is in its high precision in most queries due to its stricter matching. It accounts for review sentiment, reducing false positives. It is also context aware, which filters out unrelated mentions. The limitations of this model is that it may miss relevant matches if opinion words are paraphrased or the proximity window is too tight. It is also not based on machine learning, so it has limited adaptability. Finally, the model is slightly more computationally intensive.

M2 consistently performed best overall in precision across most queries. For image quality, it captured high quality, relevant reviews while filtering vague ones. For the mouse button, it correctly filtered out many irrelevant mentions that plagued M1 and Baseline. For wifi signal, M2 gave a huge boost in precision by enforcing the need for both a polarity match and opinion proximity.

Precision Evaluation Methodology

Precision is a fundamental metric in information retrieval that quantifies the accuracy of a system in retrieving only relevant documents. In the context of this project, precision is defined as the number of relevant reviews retrieved over the total number of reviews retrieved. Our evaluation is focused on five specific queries: audio quality - poor, wifi signal - strong, mouse button - click problem, gps map - useful, image quality - sharp. Each query is associated with a specific aspect and a target opinion polarity, positive or negative. The precision of each model, baseline, advanced m1, and advanced m2, was measured by how well it retrieved reviews that truly matched both the aspect and opinion for the query.

Each query was encoded with two aspect terms such as audio and quality, a list of opinion words such as poor, strong or sharp, and a polarity label, such as pos for positive and neg for negative. This structured format was used consistently across all models.

For Baseline model retrieval, we implemented a basic boolean match as we mentioned before. The review was included if it contained both aspect terms anywhere in the body text. These matching reviews were written to a .txt file. After processing the entire dataset, the first sixty matched reviews were selected and saved for each query.

For Advanced M1 retrieval, we searched for aspect keywords in both title and text. We did not require proximity between aspect and opinion words. As with baseline, the first sixty were selected.

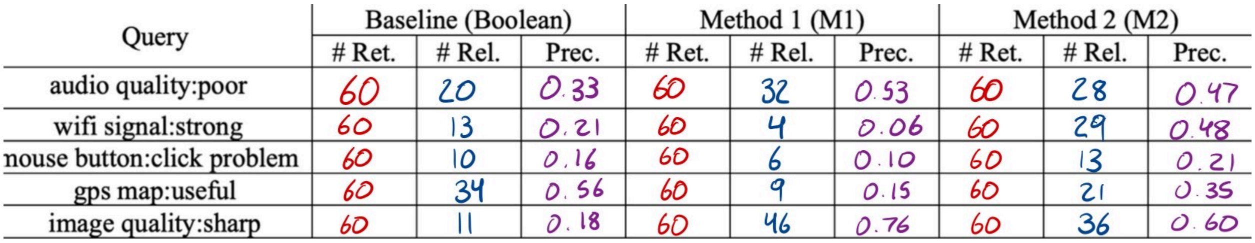
For Advanced M2 retrieval, we introduce a proximity window where the aspect and opinion words must appear close and a sentiment filter, where positive queries require ratings greater than or equal to four and negative queries require ratings less than or equal to two. Both title and bodies were scanned. This more sophisticated approach required additional logic but returned far fewer false positives.

After retrieving the top sixty results for each model query pair, each .txt file was manually labeled. For every review, we asked if it clearly mentioned the aspect terms, expressed an opinion that matches the intended polarity, and if the sentiment was about that aspect. Reviews were labeled as either relevant or irrelevant. To help with this process, each review was formatted with its review_id, title, rating, and full text. Files were organized per model and query for easy navigation. For each sixty review set, we counted how many were manually marked as relevant and applied the precision formula.

The tool we used came from the Advanced M2 model. It loaded reviews from reviews_segment.pkl, cleaned text and ratings. It also applied logic to match aspect, opinion, proximity and rating. It exported review IDs for each query. Then we exported the reviews to a .txt file, as I mentioned before, read the IDs retrieved by each model and matched them back to full reviews in the original dataset, then wrote them into human readable files with numbering, rating, title and text.

This section was the most confusing and problematic to implement. At first, I went with just counting manually the original output files that models gave. I came up with the idea to make a program to retrieve specific reviews so that the number of them can be feasible, however that task was too much. Finally, I went with the idea to create a random sample formula that retrieves a specific number of reviews. The new problem was choosing the number of samples to retrieve. If I chose a low number of samples, my confidence level would be very low and my estimation will contain bias. Therefore, the more samplings I take, the more accurate my estimation will be. Since my collected number of reviews was in the thousands, sixty to hundred samples will be sufficient. Just like I mentioned previously, I chose sixty.

Below is the table, which contains the number of reviews retrieved, the number of relevant reviews, and the precision for that query relating to the model used.



Query	Baseline (Boolean)			Method 1 (M1)			Method 2 (M2)		
	# Ret.	# Rel.	Prec.	# Ret.	# Rel.	Prec.	# Ret.	# Rel.	Prec.
audio quality:poor	60	20	0.33	60	32	0.53	60	28	0.47
wifi signal:strong	60	13	0.21	60	4	0.06	60	29	0.48
nouse button:click problem	60	10	0.16	60	6	0.10	60	13	0.21
gps map:useful	60	34	0.56	60	9	0.15	60	21	0.35
image quality:sharp	60	11	0.18	60	46	0.76	60	36	0.60

Advanced M2 consistently outperformed others, especially on queries with sentiment like mouse button or wifi signal. Baseline worked surprisingly well for broad terms like audio quality as it casts a wide net. Advanced M1 underperformed. As we can see it returned many irrelevant mentions due to lack of sentiment or proximity checks.

Discussion and Analysis

The precision results across all models and queries reveal several important patterns that reflect the design tradeoffs inherent in each approach. We will talk about the model behaviors in more depth, using specific queries and results to analyze how different retrieval techniques succeeded or failed.

Baseline Model Behavior

Despite its simplicity, the Baseline model performed surprisingly well in certain cases, particularly for broad generic queries. These queries involve terms that are often co-located in product reviews. Since the model only checks for the presence of both aspect keywords, it captures a wide net of potentially relevant reviews.

However, this strength also exposes its weakness, its lack of discrimination. For example, in the mouse button query, many irrelevant reviews were retrieved simply because they mentioned both mouse and button in different contexts. Similarly, for wifi signal, Baseline struggled to differentiate between functional feedback and general meetings. This leads to false positives, reviews that pass the keyword check but fail to express a clear opinion about the feature in question.

Advanced M1 Behavior

Advanced M1 improved coverage by extending keyword matching to both the review title and body, and was slightly more robust in some queries. It avoided missing some relevant reviews that Baseline did not capture because they occurred in the title alone.

However, M1 often performed worse than Baseline in precision because it cast a wider net without context filtering. For instance, the wifi signal query under M1 had a very low precision. Although wifi and signal might both appear, there’s no evaluation or sentiment tied to the query intent of signal strength. M1 was especially prone to such mismatches because it didn’t require any sentiment based or proximity checks. In essence, M1 gained recall but sacrificed precision, often retrieving irrelevant reviews that would’ve been filtered out by stricter logic.

Advanced M2 Behavior

Advanced M2 demonstrated the highest precision overall, especially in queries requiring sentiment and nuanced understanding. The model required co-occurrence of aspect and opinion words within a narrow window, and the review to match the intended sentiment polarity based on rating.

This dramatically reduced false positives. For example, in the mouse button, reviews with statements that were ambiguous were correctly excluded, showing the strength of M2's, proximity constraint and sentiment alignment.

However, some true positives were missed because the opinion words did not appear within the defined window or were phrased differently. This shows M2's dependence on the predefined opinion word list. While its strictness improves relevance, it also increases the chance of false negatives when users phrase their opinions differently than expected.

Query Sensitivity

Some queries highlight how different models were more or less suited depending on the type of aspect. Image quality, being a broad and frequent theme in product reviews, saw high performance from all models. Even the baseline did well here.

The mouse button and the wifi signal were more nuanced, requiring sentiment interpretation and contextual understanding. M2 performed best here, while Baseline and M1 suffered from keyword only matching.

GPs map had middle ground performance, M2 still outperformed others by filtering reviews that expressed satisfaction or dissatisfaction with navigation or usability, whereas M1 and Baseline often retrieved unrelated uses of GPS and map.

Conclusion

In this project, we developed and evaluated three models for aspect based opinion retrieval from Amazon product reviews. The Baseline model relied on simple keyword co-occurrence, the Advanced M1 model expanded the search scope, and the Advanced M2 model introduced context sensitivity through sentiment filtering and proximity constraints.

Through manual precision evaluation across five diverse queries, we found that while the Baseline and M1 models provided broad coverage, they often returned irrelevant results due to their lack of contextual awareness. In contrast, the Advanced M2 model consistently outperformed the others, particularly in sentiment driven queries, thanks to its ability to link opinion words with aspects and filter by polarity.

Ultimately, the project highlights the importance of combining syntactic matching with semantic constraints when designing effective opinion retrieval systems.