RevieWeaver: Weaving Together Review Insights by Leveraging LLMs and Semantic Similarity

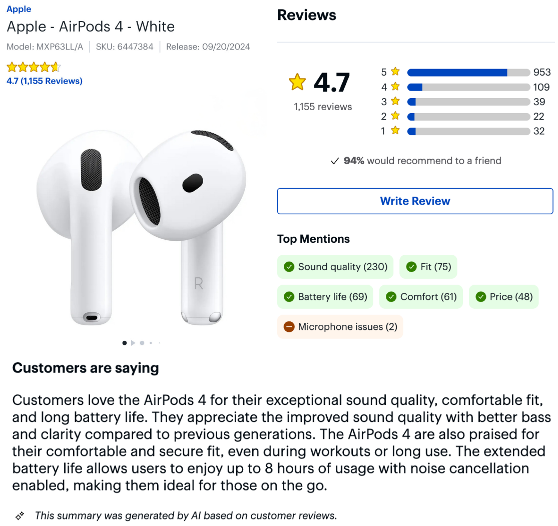
Jiban Adhikary[†], **Mohammad Alqudah**[‡*], **Arun Udayashankar**[†]
[†] Applied Machine Learning Best Buy, [‡] Microsoft
`jibankrishna.adhikary@bestbuy.com`,
`mohammad.al.qudah@hotmail.com`,
`arun.udayashankar@bestbuy.com`

Abstract

With the rise of online retail, customer reviews have become a critical factor in shaping purchasing decisions. The sheer volume of customer reviews being generated continuously presents a challenge for consumers who must sift through an overwhelming amount of feedback. To address this issue, we introduce REVIEWEAVER, a novel framework that extracts key product features and provides concise review summaries. Our innovative approach not only scales efficiently to 30 million reviews but also ensures reproducibility and controllability. Moreover, it delivers unbiased and reliable assessments of products that accurately reflect the input reviews.

1 Introduction

At Best Buy[1], a substantial number of customer reviews are collected daily, resulting in a comprehensive collection of shared experiences for each product. Over time, these reviews can accumulate to tens of thousands, providing an opportunity to uncover valuable insights into the product's strengths and weaknesses. Research shows that customer reviews significantly influence purchasing decisions (Li et al., 2020). During the shopping experience, customers can examine a set of reviews left by previous customers, allowing them to gain a deeper understanding of the product's features and drawbacks. However, when a product has an excessive amount of reviews, this process can become overwhelming. Providing a condensed list of a product's key features, pros, and cons, along with a brief summary of customer opinions can help mitigate this issue. This approach enables customers to quickly and efficiently assess the product's strengths and weaknesses, without being bogged down by an excessive amount of information.



Apple
Apple - AirPods 4 - White
Model: MXP63LL/A | SKU: 6447384 | Release: 09/20/2024
4.7 (1,155 Reviews)

Reviews

4.7
1,155 reviews

5 ★	953
4 ★	109
3 ★	39
2 ★	22
1 ★	32

✓ **94%** would recommend to a friend

Top Mentions

✓ Sound quality (230)
✓ Fit (75)
✓ Battery life (69)
✓ Comfort (61)
✓ Price (48)
⊖ Microphone issues (2)

Customers are saying

Customers love the AirPods 4 for their exceptional sound quality, comfortable fit, and long battery life. They appreciate the improved sound quality with better bass and clarity compared to previous generations. The AirPods 4 are also praised for their comfortable and secure fit, even during workouts or long use. The extended battery life allows users to enjoy up to 8 hours of usage with noise cancellation enabled, making them ideal for those on the go.

This summary was generated by AI based on customer reviews.

Figure 1: Review Distillation and Summarization of product reviews in Best Buy.

1.1 Contributions

In this paper, we propose a unified and scalable solution to extract a product's key features from customer reviews and then use the extracted features to generate a concise summary. The process of extracting the essential features from customer reviews will henceforth be referred to as *review distillation*. For review distillation and review summarization, we utilize a range of methodologies and strategies. At present, large language models (LLMs) such as ChatGPT, GPT-4, GPT-4o, Llama, and Gemini are widely employed to tackle numerous natural language tasks. As such, review distillation and review summarization tasks can also be solved using an LLM. These LLMs have a larger context size (2K–1M tokens) and theoretically thousands of reviews can be passed to them for distillation and summarization. However, using all the reviews as context is not ideal due to factors such as cost, re-usability, reproducibility, controllability, or scalability. Our framework also employs an LLM, but with a more judicious use of context, taking

[*] Work done while the author was employed at Best Buy.
[1] `https://www.bestbuy.com`

Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Industry Track), pages 431–448
April 30, 2025 ©2025 Association for Computational Linguistics

these factors into account.

We make the following four contributions:

1. We present a comprehensive and scalable framework for review distillation, which involves extracting pros and cons from millions of customer reviews. Our method addresses the challenges of implicit aspect extraction and utilizes LLMs to facilitate the process.

2. To further enhance the review distillation process, we leverage a classic union-find algorithm (Galler and Fisher, 1964) and utilize union-by-rank and semantic similarity to facilitate the extraction of meaningful features.

3. We expand our framework to generate a comprehensive and accurate summary of reviews utilizing an LLM and a curated set of essential features and customer testimonials, thereby ensuring reproducibility and fairness while avoiding the use of excessive context.

4. We make the source code and a review dataset publicly available for future research[2].

2 Related Work

2.1 Aspect based sentiment analysis

Sentiment Analysis (SA) is one of the frequently studied topics in the field of Natural Language Processing (NLP). Generally, SA can be performed at three levels: document-level, sentence-level, and aspect-level. Aspect based sentiment analysis aims to extract aspects from textual chunks and assign sentiments to them. Aspect extraction (AE) can be further divided into explicit and implicit categories. Explicit aspects are explicitly mentioned in the text, such as *drawers* in the review “the refrigerator has spacious drawers”. In contrast, implicit aspects are not explicitly stated but can be inferred from the text, like *battery life* in the statement “the phone cannot last a full day of use”.

The process of AE remains challenging, and various methodologies have been employed to extract aspects from text. Amazon has a solution to extract aspects and sentiments from customer reviews [3], but it was not disclosed how the solution was implemented and scaled. Researchers have leveraged textual sequences using Recurrent Neural Networks (RNNs) (Wang et al., 2016) such as BiLSTM and CRF (Giannakopoulos et al., 2017), as well as hierarchical multi-layer Bidirectional Gated Recurrent Units (BiGRUs) (Ma et al., 2018). These models can be trained in either supervised or unsupervised manners. Additionally, attention mechanisms have been incorporated (Liu et al., 2015; Li and Lam, 2017; He et al., 2017) to enhance the capture of relationships between aspects and their corresponding sentiments. While Sentiment Analysis (SA) can be performed separately from AE, many recent approaches combine these processes into a single pipeline. Still, existing methods face a lot of limitations, including identifying implicit aspects, handling complex sentence structures, domain-specificity, reliance on labeled data, and struggles with ambiguous language (Mughal et al., 2024; Ahmed et al., 2023; Chifu and Fournier, 2023; Nath and Dwivedi, 2024; Wu et al., 2023; Shi et al., 2023; Yang et al., 2023).

2.2 Topic Modeling

Topic modeling aims to uncover the underlying themes within a collection of documents, with the goal of highlighting the most significant information within the document set. This process is typically performed without predefining the topics, which can lead to challenges in terms of coherence and coverage during the discovery process. In some cases, such as consumer reviews, it is important to identify both negative and positive topics. One of the earliest techniques for topic modeling is Latent Dirichlet Allocation (LDA) (Blei et al., 2003), a generative probabilistic model that assumes each document is a combination of a small number of topics, and each topic is characterized by a distribution over words. Another approach is Non-negative Matrix Factorization (NMF) (Lee and Seung, 2000), a mathematical technique that decomposes a matrix containing only nonnegative values into two new matrices. By multiplying these matrices together, the original matrix can be reconstructed, allowing for the extraction of topics from a large document-word matrix. While LDA and NMF are computationally intensive, recent advances have incorporated textual embeddings into topic discovery. These embeddings are created, then by using distance measures in an embedding space the embeddings are aggregated using methods such as K-means. Word2Vec was used in (Qiang et al., 2017) to create the embeddings for discovering topics, while more recent approaches

[2] https://github.com/sworborno/RevieWeaver

[3] https://www.aboutamazon.com/news/amazon-ai/amazon-improves-customer-reviews-with-generative-ai

have utilized variants of BERT (Devlin et al., 2019), such as Top2Vec (Angelov and Inkpen, 2024) and BERTopic (Grootendorst, 2022), to create the embeddings. Large language models (LLMs) have also shown promise in topic modeling (Wang et al., 2024), with LLMs like GPT being prompted to extract topics from text corpora.

2.3 Summarization

Text summarization is the process of condensing a source text into a shorter version while preserving its essential information and meaning. This task is particularly crucial in consumer reviews, where opinion summaries are frequently extracted. There are two primary techniques for opinion summaries: non-textual summaries, such as aggregated ratings, aspect-sentiment tables, and opinion clusters; and textual summaries, which often involve extracting a brief text from the original reviews. Textual summarization can be accomplished through either abstractive or extractive methods. In the context of customer reviews, abstractive summarization is more beneficial due to the vast amount of text and diverse range of opinions (Kim Amplayo et al., 2022). Recent advancements in deep learning and pre-trained language models like BERT, T5 (Raffel et al., 2020), and other models have significantly improved abstractive summarization (Ramina et al., 2020). Hybrid approaches that combine elements of both techniques can also enhance summary quality. Furthermore, the integration of large language models (LLMs) has pushed the field forward, enabling the generation of high-quality summaries.

2.4 Challenges of opinion mining

We address several challenges in this work, particularly in the realm of implicit aspects, which are less well-studied due to the lack of clarity in identifying them. Unlike explicit aspects, sentences often do not contain explicit names or clues for the extracted aspects. Moreover, implicit aspect extraction has practical applications in customer reviews, as demonstrated by Nazir et al. (2020). In this work, we use an LLM as a zero-shot model to overcome the complexity of extracting implicit aspects. In addition, we show a methodology to overcome the coherence challenges in topic discovery within customer reviews, where the topics (pros or cons) are hidden within a skewed dataset, where for example, finding cons in an overwhelming number of positive reviews can be challenging. Lastly, there are several challenges when producing review summaries. First, scalability is critical to handling a large volume of input reviews, requiring the ability to retrieve implicit insights at scale. Secondly, faithfulness guarantees that the summary accurately mirrors the input reviews, avoiding any confusion of entities or disregarding entities mentioned by only one or two customers. Finally, controllability allows for the creation of constrained summaries, avoiding problems such as focusing solely on positive opinions and unintentionally leaving out negative opinions in product reviews. Our work addresses these challenges.

3 Problem Statement

Let $R = \{r_1, r_2, ..., r_n\}$ be a set of customer reviews for a product P, where each review r_i is a sequence of words. We have mainly two tasks:

(i) Review distillation: Extract a set of features $F = \{F^+,\ F^-\}$, where each feature $f_k^+ \in F^+$ is a phrase that represents a positive feature and $f_l^- \in F^-$ is a phrase that represents a negative feature. We further formulate this task into two sub-tasks:

(a) Aspect-sentiment extraction: Given a review $r_i \in R$, identify a set of tuples (a_j, e_j, q_j), where a_j is an aspect that expresses a sentiment (positive or negative) e_j towards the product and q_j is a representative quote.

(b) Aspect grouping: Group the identified tuples into two sets of features based on their semantic similarity: positive features $f_k^+ \in F^+$ and negative features $f_l^- \in F^-$. Each positive and negative feature has also a set of representative quotes, q_k^+ and q_k^-, respectively.

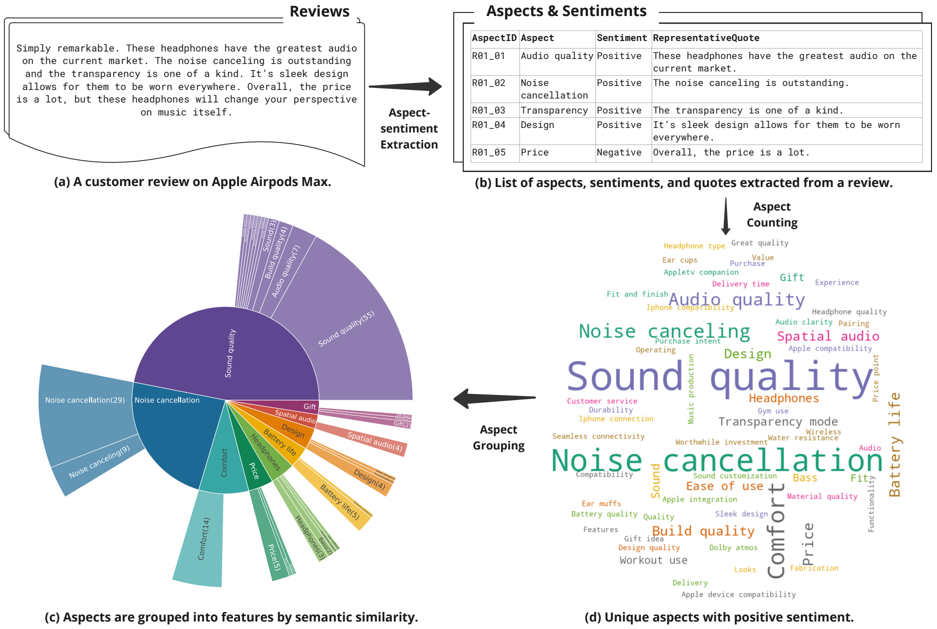
(ii) Review summarization: Generate a concise and informative summary S that captures the key sentiments and insights expressed in reviews, R.

4 Approach

We propose a unified framework named REVIEWEAVER to extract high-level product features from customer reviews and generate a concise and helpful summary of the reviews.

4.1 Aspect-sentiment extraction

We choose to extract aspects and sentiments using the review text and an LLM. For a given review, we prompt the LLM to extract top five aspects, the associated sentiments, and representative quotes. Our



Simply remarkable. These headphones have the greatest audio on the current market. The noise canceling is outstanding and the transparency is one of a kind. It's sleek design allows for them to be worn everywhere. Overall, the price is a lot, but these headphones will change your perspective on music itself.

(a) A customer review on Apple Airpods Max.

AspectID	Aspect	Sentiment	RepresentativeQuote
R01_01	Audio quality	Positive	These headphones have the greatest audio on the current market.
R01_02	Noise cancellation	Positive	The noise canceling is outstanding.
R01_03	Transparency	Positive	The transparency is one of a kind.
R01_04	Design	Positive	It's sleek design allows for them to be worn everywhere.
R01_05	Price	Negative	Overall, the price is a lot.

(b) List of aspects, sentiments, and quotes extracted from a review.

(c) Aspects are grouped into features by semantic similarity.

(d) Unique aspects with positive sentiment.

Figure 2: With review distillation, for each customer review, we find a list of aspects, their associated sentiments, and representative quotes in the review, illustrated in Figures (a) and (b). Next, we categorize these aspects into two groups based on their sentiment. For example, Figure (d) highlights the positive aspects of the Apple Airpods Max. The larger font sizes indicate higher frequency of mention for each aspect. Finally, we group similar aspects together based on their semantic similarity, as seen in Figure (c), where each cluster is labeled with the most frequently mentioned aspect and referred to as a *feature*. Note, only features with a count of three or more are displayed.

rationale for extracting the representative quotes is twofold: firstly, we leverage the representative quotes to calculate an average text embedding for each distinct aspect and secondly, we employ the quotes while generating summaries of the reviews.

4.2 Aspect grouping

After we find the tuples (aspect, sentiment, and representative quote) for all the reviews of a product, we categorize the tuples based on their sentiments, with each sentiment comprising a list of aspects. For each sentiment, we combine the unique aspects to create a "bag-of-aspects" and count how many times they have been mentioned in the reviews. In this case, aspects like *easy to use* and *ease of use* are considered completely unique. For each unique aspect, we also keep a list with all the representative quotes of that aspect. The size of the list is usually equal to the number of mentions. Then we use a clustering algorithm to find and merge similar aspects. We denote each cluster as a *feature*. For instance, the aspects *easy to use*, *easy setup*, and *convenient* could be termed as the feature *easy to use*. Figure 2 illustrates the steps involved in review distillation.

4.3 Summarization

Following the meticulous review distillation process, we obtain two distinct lists: one comprising the product's positive features and the other having its negative features. Each feature is accompanied by a collection of relevant quotes. To facilitate the generation of a concise summary, we employ an LLM and present it with the top 10 positive and top 10 negative features, along with each feature's top 10 representative quotes. This approach enables us to circumvent the need to provide the entirety of the reviews as context for the LLM. Additionally, we instruct the LLM to initiate the summary with a random phrase from a predetermined list (Table 9), thereby ensuring the opening sentence of the summary varies across different products.

5 Experiments

5.1 Dataset

To assess the effectiveness of our proposed framework, we compiled a dataset based on reviews received on our online platform for various products. Due to the large volume of reviews, we selected a representative sample of reviews. Each review submitted on our platform undergoes a thorough moderation process prior to publication. Reviews

Technique	Silhouette coefficient↑		Calinski-Harabasz index↑		Davies-Bouldin index ↓	
	top-5	top-10	top-5	top-10	top-5	top-10
DBSCAN	0.31 ± 0.18	0.35 ± 0.18	10.26 ± 23.28	8.71 ± 18.08	1.09 ± 0.19	1.07 ± 0.17
HDBSCAN	0.43 ± 0.18	0.44 ± 0.17	14.18 ± 37.69	11.29 ± 27.74	1.39 ± 0.34	1.35 ± 0.29
REVIEWEAVER	**0.59** ± 0.17	**0.52** ± 0.16	**19.99** ± 34.04	**13.14** ± 18.14	**0.65** ± 0.30	**0.58** ± 0.25

Table 1: Results for different clustering techniques. Results formatted as: $mean \pm SD$. ↑ indicates more is better, ↓ indicates less is better.

containing personal information, explicit language, fraudulent content, or harmful material are not accepted and are rejected. Here, we only selected reviews that had already been deemed appropriate for publication.

We chose the best-selling products within the last 30 days prior to the writing of this paper. Each product had a minimum of 2 and a maximum of 78,000 reviews, and we randomly selected one percent of these reviews for each product. If the sample size was less than 15, we excluded the product from the dataset. Our final dataset consists of 167 products and 10,103 reviews. Each review has on average 28 tokens and 103 billable characters. The number of tokens and billable characters was determined by the LLM tokenizer.

5.2 Review Distillation

Prompting. For each review in our dataset, we used a prompt (Figure 4) and assigned an LLM with extracting aspects, sentiments, and representative quotes. We used Google `gemini-1.5-flash` for this task. This model was chosen due to its cost-effectiveness and alignment with the company's policy. To streamline the process, we utilized a batch process when making LLM calls, with batch sizes ranging from 5 to 10 based on the length of the reviews. We prompted the LLM to produce structured output (JSON format).

Clustering. After extracting the aspects from the reviews, we separated the aspects with positive sentiments from those with negative sentiments. For each group, we identified unique aspects and their corresponding counts. We then applied clustering algorithms to group similar aspects. Our clustering methods included a union-find algorithm (Galler and Fisher, 1964) with rank and semantic similarity, and two unsupervised clustering algorithms, namely DBSCAN (Ester et al., 1996) and HDBSCAN (Campello et al., 2013).

Union-find by ranking & similarity. We refined the traditional union-find algorithm for disjoint data structures by adapting it to group semantically similar aspects. Each aspect was represented as an independent node in a graph, and we assumed that two nodes would form a cluster if they shared similar semantic meaning. To facilitate this process, each node was assigned a unique identifier, the name of the aspect, a mention count or ranking, a list of representative quotes, and a parent identifier. Initially, the parent identifier for each node was the same as its node identifier. Additionally, we precomputed two embeddings for each node: (1) an aspect embedding, which represented the semantic meaning of the aspect's name, and (2) a quote embedding, which was an average embedding of the representative quotes. We utilized the sentence transformer (Reimers and Gurevych, 2019) and a pre-trained `all-MiniLM-L6-v2` model with a batch size of 192 to compute these embeddings. During the union of two nodes (Algorithm 3), we compared their aspect embeddings and quote embeddings using cosine similarity. If similarities exceeded a predetermined threshold, we merged the nodes. In this case, the node with the higher mention count became the parent node, and all attributes of the child node was attributed to the parent node. The specific modifications are detailed in Algorithm 4.

5.2.1 Evaluation

On the extracted aspect, sentiment, and representative quote tuples, we applied the modified union-find algorithm, DBSCAN, and HDBSCAN. For DBSCAN and HDBSCAN, we computed embeddings for each aspect and utilized them as model features. The specific parameters and values for these models are shown in Appendix A.3. Due to the lack of ground truth labels, we assessed the clustering algorithms using three appropriate techniques for unsupervised clustering: the Silhouette coefficient (Rousseeuw, 1987), the Calinski-Harabasz index (Caliński and Harabasz, 1974), and the Davies-Bouldin index (Davies and Bouldin, 1979). Furthermore, as the three algorithms did not produce the same number of clusters, we examined the top-5 and top-10 clusters from each

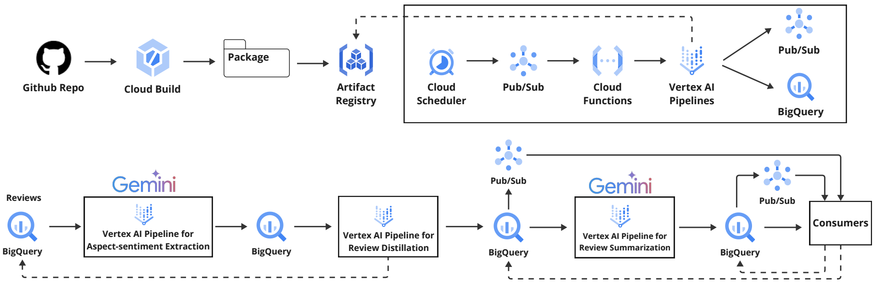


Figure 3: Deployment pipelines of REVIEWEAVER.

Criteria	full-context	distilled-context
Coherence	4.22 ± 0.41	4.14 ± 0.41
Consistency	4.32 ± 0.47	4.28 ± 0.58
Fluency	4.76 ± 0.43	4.69 ± 0.46
Relevance	4.13 ± 0.35	4.07 ± 0.49

Table 2: Evaluation results on generated summaries. Results formatted as $mean \pm SD$.

method for comparison. The results are presented in Table 1. It reveals that the modified union-find algorithm in REVIEWEAVER achieved the most optimal scores, indicating its superiority over DBSCAN and HDBSCAN.

5.3 Review Summarization

We conducted experiments to create a high-level summary of customer reviews for a given product. To avoid utilizing all reviews, we leveraged the extracted features from review distillation. For each set of positive and negative features (pros and cons), we collected the feature names, mention counts, and up to 10 representative quotes discussing a feature. When there were more than 10 quotes for a particular feature, we employed a priority queue with a set of heuristics to determine the top 10 quotes. These heuristics included the number of characters or words in each quote and the presence of the feature or aspect in the quote. We crafted a prompt (Figure 5) encompassing the product name, its pros and cons, and the associated mention counts, and asked Google `gemini-1.5-flash` to generate a summary.

While the main purpose of our summarization task was to use a condensed set of information, for comparison, we also generated summaries for all the products in our dataset using the full set of available reviews. We used the same prompt mentioned above except we switched the content to use all available reviews (Figure 6).

5.3.1 Evaluation

To assess the quality of the summaries produced using various context types, we employed a language model (LLM) as a judge based on several criteria. We adhered to the four evaluation metrics outlined by Fabbri et al. (2021) and Liu et al. (2023): coherence, consistency, fluency, and relevance. For each criterion, we adapted the prompts (Figure 7, 8, 9, 10) presented in Liu et al. (2023) and requested Google's `gemini-1.5-pro` to evaluate the summaries on a scale of 1 to 5, where 1 is the lowest and 5 is the highest. The mean and standard deviation of the scores are displayed in Table 2. For each criterion, we performed the Wilcoxon signed-rank test and the Mann-Whitney U-Test (Table 8), which revealed no significant differences between summaries created with full context and those generated with distilled context, indicating that the summaries produced with the distilled features are comparable to those produced with all reviews. See Table 10 for some sample summaries.

6 Deployment

At Best Buy, we utilize Google Cloud to host our data analytics and machine learning operations. Figure 3 shows the deployment pipelines used to run REVIEWEAVER. To execute the proposed framework, we package REVIEWEAVER as a Python package to be executed on multiple cloud instances, as illustrated in the top section. We then leverage a series of Google Cloud services to schedule and trigger pipelines, which employ the built package to process the reviews and produce the final output for customer display. This strategic approach enables us to decouple our code, deployment, and hardware, allowing us to utilize the same infrastructure for both experimental and large-scale production runs. To date, our framework has successfully processed approximately 30 million reviews across a staggering 200,000 product categories, demonstrating its robustness and scalability.

7 Discussion

One of our focuses in this work was to ensure that the produced data could be extracted effectively at scale, and to ensure that we produce fair and controllable review distillation and summaries. Scalability was achieved by decoupling the aspect extraction. When the LLM is used, the data is cached for later use. The grouping and ranking steps can be run multiple times without the need to re-run the costly aspect extraction step. For new incoming reviews, continuous updates will also cost less since new reviews will be processed once. As a result, the long-term costs for aspect extraction will be capped.

For the review summarization process, we effectively reduced the number of input tokens and, consequently, the associated cost for summary generation using the LLM. Since we use at most the top 10 positive features, the top 10 negative features, and the top 10 representative sentences for each feature, the upper limit of context size will always be capped at a certain number of tokens irrespective of the total number of reviews. This significantly reduced the cost of summarizing the content of products that have thousands of reviews.

One limitation of our work is that we only used a single model to evaluate the summaries, primarily due to enterprise policies and privacy concerns. However, we believe that using multiple models would have yielded similar judgments.

Controllability is crucial in industrial settings, since such systems are semi-autonomous and we cannot manually review each output. We have seen that our approach produces repeatable outputs across diverse product categories. Lastly, as a retailer, its our responsibility to surface unbiased and fair information to the customer, and let them use it to aid their purchasing decision. Using REVIEWEAVER, we ensured that both pros and cons are adequately represented in both review distillation and product summaries.

8 Conclusion

We have shown that REVIEWEAVER addresses some of the main challenges in review distillation and summarization. In our experiments and real-world application, we saw that REVIEWEAVER outperforms other methods both in empirical metrics and in reproducibility, cost effectiveness, and fairness.

Acknowledgments

We would like to express our sincere gratitude to Erinn Swinton, Jeffrey Prachick, Peter Jentz, Ankush Gupta and other members of the User Generated Content team for painstakingly reviewing the generated distillations and summaries and providing invaluable feedback.

References

Kanwal Ahmed, Muhammad Imran Nadeem, Zhiyun Zheng, Dun Li, Inam Ullah, Muhammad Assam, Yazeed Yasin Ghadi, and Heba G Mohamed. 2023. Breaking down linguistic complexities: A structured approach to aspect-based sentiment analysis. *Journal of King Saud University-Computer and Information Sciences*, 35(8):101651.

Dimo Angelov and Diana Inkpen. 2024. Topic modeling: Contextual token embeddings are all you need. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 13528–13539, Miami, Florida, USA. Association for Computational Linguistics.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3(null):993–1022.

Tadeusz Caliński and Jerzy Harabasz. 1974. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27.

Ricardo JGB Campello, Davoud Moulavi, and Jörg Sander. 2013. Density-based clustering based on hierarchical density estimates. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 160–172. Springer.

Adrian-Gabriel Chifu and Sébastien Fournier. 2023. Sentiment difficulty in aspect-based sentiment analysis. *Mathematics*, 11(22).

David L Davies and Donald W Bouldin. 1979. A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence*, PAMI-1(2):224–227.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, page 226–231. AAAI Press.

Alexander R Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. Summeval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409.

Bernard A Galler and Michael J Fisher. 1964. An improved equivalence algorithm. *Communications of the ACM*, 7(5):301–303.

Athanasios Giannakopoulos, Claudiu Musat, Andreea Hossmann, and Michael Baeriswyl. 2017. Unsupervised aspect term extraction with B-LSTM & CRF using automatically labelled datasets. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 180–188, Copenhagen, Denmark. Association for Computational Linguistics.

Maarten Grootendorst. 2022. BERTopic: Neural topic modeling with a class-based TF-IDF procedure. *arXiv preprint arXiv:2203.05794*.

Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2017. An unsupervised neural attention model for aspect extraction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 388–397.

Reinald Kim Amplayo, Arthur Brazinskas, Yoshi Suhara, Xiaolan Wang, and Bing Liu. 2022. Beyond opinion mining: Summarizing opinions of customer reviews. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3447–3450.

Daniel Lee and H Sebastian Seung. 2000. Algorithms for non-negative matrix factorization. *Advances in neural information processing systems*, 13.

Kunlin Li, Yuhan Chen, and Liyi Zhang. 2020. Exploring the influence of online reviews and motivating factors on sales: A meta-analytic study and the moderating role of product category. *Journal of Retailing and Consumer Services*, 55:102107.

Xin Li and Wai Lam. 2017. Deep multi-task learning for aspect term extraction with memory interaction. In *Proceedings of the 2017 conference on empirical methods in natural language processing*, pages 2886–2892.

Pengfei Liu, Shafiq Joty, and Helen Meng. 2015. Fine-grained opinion mining with recurrent neural networks and word embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1433–1443, Lisbon, Portugal. Association for Computational Linguistics.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: NLG evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore. Association for Computational Linguistics.

Dehong Ma, Sujian Li, and Houfeng Wang. 2018. Joint learning for targeted sentiment analysis. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4737–4742, Brussels, Belgium. Association for Computational Linguistics.

Nimra Mughal, Ghulam Mujtaba, Sarang Shaikh, Aveenash Kumar, and Sher Muhammad Daudpota. 2024. Comparative analysis of deep natural networks and large language models for aspect-based sentiment analysis. *IEEE Access*, 12:60943–60959.

Deena Nath and Sanjay K Dwivedi. 2024. Aspect-based sentiment analysis: approaches, applications, challenges and trends. *Knowledge and Information Systems*, 66(12):7261–7303.

Ambreen Nazir, Yuan Rao, Lianwei Wu, and Ling Sun. 2020. Issues and challenges of aspect-based sentiment analysis: A comprehensive survey. *IEEE Transactions on Affective Computing*, 13(2):845–863.

Jipeng Qiang, Ping Chen, Tong Wang, and Xindong Wu. 2017. Topic modeling over short texts by incorporating word embeddings. In *Advances in Knowledge Discovery and Data Mining: 21st Pacific-Asia Conference, PAKDD 2017, Jeju, South Korea, May 23-26, 2017, Proceedings, Part II 21*, pages 363–374. Springer.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Mayank Ramina, Nihar Darnay, Chirag Ludbe, and Ajay Dhruv. 2020. Topic level summary generation using bert induced abstractive summarization model. In *2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS)*, pages 747–752. IEEE.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Peter J. Rousseeuw. 1987. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65.

Jingli Shi, Weihua Li, Quan Bai, Yi Yang, and Jianhua Jiang. 2023. Syntax-enhanced aspect-based sentiment analysis with multi-layer attention. *Neurocomputing*, 557:126730.

Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. 2016. Recursive neural conditional random fields for aspect-based sentiment analysis. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 616–626, Austin, Texas. Association for Computational Linguistics.

Xinyi Wang, Wanrong Zhu, Michael Saxon, Mark Steyvers, and William Yang Wang. 2024. Large language models are latent variable models: explaining and finding good demonstrations for in-context learning. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Haiyan Wu, Chaogeng Huang, and Shengchun Deng. 2023. Improving aspect-based sentiment analysis with knowledge-aware dependency graph network. *Information Fusion*, 92:289–299.

Heng Yang, Chen Zhang, and Ke Li. 2023. PyABSA: A modularized framework for reproducible aspect-based sentiment analysis. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, CIKM 2023, Birmingham, United Kingdom, October 21-25, 2023*, pages 5117–5122. ACM.

A Appendix

A.1 LLM as aspect-sentiment extractor

In Section 5.2, we used an LLM to extract triplets comprising aspect, sentiment, and a representative quote from the reviews. Utilizing Google `gemini-1.5-flash` as a zero-shot model, we bypassed the traditional multi-step pipeline of Aspect-Based Sentiment Analysis (ABSA), which typically involves entity recognition, aspect identification, and sentiment analysis. As previously discussed, existing ABSA models face challenges in discerning implicit aspects within reviews. Furthermore, the identified aspects often consist of verbatim word matches from the text, resulting in potentially inaccurate or insufficiently descriptive aspect representations. For instance, in the sentence "This is a must-buy product, the sound is great", a conventional ABSA model might extract 'sound' as the aspect. However, "sound quality" would be a more appropriate and informative aspect in this context.

Our empirical findings demonstrate that leveraging an LLM effectively addresses these limitations. Applying our methodology to the experimental dataset yielded 17,331 tuples of aspect, sentiment, and quote. Notably, only 491 (2.83%) of the extracted aspects were exact word matches from the source text. The remaining aspects were either implicit, or automatically generated with meaningful and contextually relevant wording. See Table 3 for some examples.

Aspect	Representative Quote
Portability	They are very convenient to use on the go.
Value	You really get the bang for your buck!
Charging speed	Usually charge quickly.
Battery life	It stopped keeping charge as it used to in the beginning.
Connectivity	The syncing would be funky at times.
Sound quality	The sound is great!
Compatibility	Easily integrated with iPhone and iPad.
Noise isolation	It does not prevent outside noise.
Call quality	Super convenient to take calls with.
Durability	Great earphones that last long.
Reaction time	Quick reaction during gameplay.
Haptic feedback	Unbelievable feedback from this controller.
Leakproof	Very flexible, durable, and do not leak.
Affordability	Very affordable and worth the price.

Table 3: A list of extracted aspects and representative quotes where aspects are implicit or generated with meaningful and contextually relevant wording.

A.2 Additional deployment details

Our deployment process for ReviewWeaver consists of three Vertex AI pipelines (Figure 3) on Google Cloud Platform: (i) aspect-sentiment extraction pipeline, (ii) review distillation pipeline, and (iii) review summarization pipeline. The aspect-sentiment extraction pipeline runs on a daily schedule and processes the moderated reviews that have become available on our data platform within the last 24 hours. To ensure efficient processing, we batch the reviews and make parallel LLM calls to extract the aspects, sentiments, and quotes from each review. Additionally, we have implemented rate limiters to prevent the pipeline from exceeding the quota allocated per minute. In the end, the extracted attributes are stored in a BigQuery table.

We run our review distillation pipeline on a weekly schedule, in which we process all reviews extracted via our aspect-sentiment extraction pipeline within the previous seven days. Our pipeline assesses the product categories and determines whether we have previously identified positive and negative features for a product or if we need to conduct a fresh analysis. For new products or reviews, we employ Algorithm 4 to identify the relevant positive and negative features.

In contrast, for existing products and new reviews, we first calculate the number of delta reviews and determine whether we must adapt the existing features to accommodate the new aspects or rediscover the features entirely. If the number of delta reviews exceeds 50% of the total reviews, we re-run Algorithm 4. Otherwise, we perform similarity matching between the new aspects and existing features, merging them if the similarity threshold is met. The updated or newly discovered positive and negative features are then stored in a separate BigQuery table, notifying consumers for further processing and display on the website.

Upon completion of the review distillation pipeline, a trigger is sent to initiate the review summarization process. This pipeline examines products with newly introduced or updated features, gathers relevant information, and starts the summarization process. Once all summaries have been generated, they are uploaded to a BigQuery table and the consumers are notified to make the summaries available online.

With the aforementioned design, aspect extraction is conducted independently, and customer reviews need only be run once throughout their lifetime. This allows for the experimentation of review distillation pipelines using various similarity thresholds, and the fine-tuning of an optimal threshold that suits most products. Furthermore, the outputs from both the initial and secondary pipelines are utilized by other processes, specifically search and conversational AI, to enhance product retrieval and respond to user queries.

A.3 Clustering algorithm parameters

Parameter name	Value
eps	0.2
$min\_samples$	2
$metric$	`cosine`

Table 4: DBSCAN model parameters.

Parameter name	Value
$min\_samples$	2
$min\_cluster\_size$	2
$metric$	`cosine`
$cluster\_selection\_epsilon$	0.2

Table 5: HDBSCAN model parameters.

A.4 LLM parameters

We used Google `gemini-1.5-flash` for the aspect-sentiment extraction task in Section 5.2 and generating the summaries in Section 5.3. The model parameters and values for `gemini-1.5-flash` is listed in Table 6. We used a temperature close to zero and from our observation it did not have any significant effect on the outcomes of the model. For evaluating the summaries in Section 5.3.1, we used Google `gemini-1.5-pro`. The parameters and values of this LLM is shown in Table 7.

Parameter name	Value
$temperature$	0.01
top_p	0.80
top_k	40

Table 6: Model parameters for `gemini-1.5-flash`.

A.5 Prompts

The prompt that we used for extracting aspect, sentiment, and representative quote in Section 5.2 is shown in Figure 4. On the other hand, Figure 5 shows the prompt we used to generate summaries using distilled content and Figure 6 shows the prompt we used to generate summaries using all available reviews for a product. Figure 7, 8, 9, and 10 show the prompts we used to ask an LLM to rate the summaries based on the criteria: `coherence`, `consistency`, `fluency`, and `relevance`, respectively.

A.6 Costs of LLM calls

The costs of making LLM calls were covered through an enterprise pricing package. As of the time of writing, under a pay-as-you-go package `gemini-1.5-flash` was priced at $0.01875 per one million input characters and $0.075 per one million output characters (`https://cloud.google.com/vertex-ai/generative-ai/pricing`). In comparison, `gemini-1.5-pro` was priced at $0.3125 per one million input characters and $1.25 per one million output characters.

Parameter name	Value
$temperature$	0
top_p	0.90
top_k	40

Table 7: Model parameters for `gemini-1.5-pro`.

Algorithm 1: FIND

```
Input: $G, u$
Output: $p$
$p \leftarrow G[u].parent$
while $p \neq G[p].parent$ do
    /* Find by path compression */
    $G[p].parent \leftarrow G[G[p].parent].parent$
    $p \leftarrow G[p].parent$
$return\ p$
```

Algorithm 2: BUILD-GRAPH

```
Input: $A[(i_1, aspect_1, count_1, quotes_1[q_{11}, ..., q_{1k}]), ..., (i_n, aspect_n, count_n, quotes_n[q_{n1}, ..., q_{nl}])]$
Output: $G$
/* The following two embedding calculations were performed with a batch job */
$A[embedding_i]_{\{i=1...n\}} \leftarrow Calculate\ embedding\ of\ A[aspect_i]_{\{i=1...n\}}$
$A[quote\_embedding_i]_{\{i=1...n\}} \leftarrow Calculate\ mean\ embedding\ of\ A[quotes_{i[...]}]_{\{i=1...n\}}$
$G \leftarrow [\,]$
for each id i, aspect a, count c, quotes q, embedding e, quote_embedding qe in $A$ do
    $N \leftarrow \varnothing$
    $N.parent \leftarrow i$
    $N.name \leftarrow a$
    $N.rank \leftarrow c$
    $N.quotes \leftarrow q$
    $N.embedding \leftarrow e$
    $N.quote\_embedding \leftarrow qe$
    $N.other\_names \leftarrow \{name\}$
    $G[i] \leftarrow N$
$return\ G$
```

Algorithm 3: UNION

```
Input: $G, u, v$
Output: $No\ output,\ modifies\ the\ graph\ nodes$
$p_1 \leftarrow FIND(G, u)$ /* Call Algorithm 1 */
$p_2 \leftarrow FIND(G, v)$ /* Call Algorithm 1 */
if $p_1 = p_2$ then
    $return$
$name_1 \leftarrow G[p1].name$
$name_2 \leftarrow G[p2].name$

$emb_1 \leftarrow G[p_1].embedding$
$emb_2 \leftarrow G[p_2].embedding$
$sembed_1 \leftarrow G[p_1].quote\_embedding$
$sembed_2 \leftarrow G[p_2].quote\_embedding$

$similarity \leftarrow COSINE - SIMILARITY(emb_1, emb_2)$
$sent\_similarity \leftarrow COSINE - SIMILARITY(sembed_1, sembed_2)$
/* Check if calculated similarities are greater than predefined thresholds */
/* Thresholds used: SIMILARITY = 0.50, SENTENCE_SIMILARITY = 0.40 */
if $similarity \geq SIMILARITY\ \&\ sent\_similarity \geq SENTENCE\_SIMILARITY$ then
    if $G[p_1].rank = G[p_2].rank$ then
        $len_1 \leftarrow LENGTH(name_1)$/* Get the number of characters in $name_1$ */
        $len_2 \leftarrow LENGTH(name_2)$/* Get the number of characters in $name_2$ */
        if $len_1 \leq len_2$ then
            /* Pick the node with the shorter name as parent */
            $G[p_2].parent \leftarrow p_1$
            $G[p_1].rank \leftarrow G[p_1].rank + G[p_2].rank$
            $G[p_1].quotes.update(G[p_2].quotes)$
            $G[p_1].other\_names.update(G[p_2].other\_names)$
        else
            $G[p_1].parent \leftarrow p_2$
            $G[p_2].rank \leftarrow G[p_2].rank + G[p_1].rank$
            $G[p_2].quotes.update(G[p_1].quotes)$
            $G[p_2].other\_names.update(G[p_1].other\_names)$
    else if $G[p_1].rank > G[p_2].rank$ then
        $G[p_2].parent \leftarrow p_1$
        $G[p_1].rank \leftarrow G[p_1].rank + G[p_2].rank$
        $G[p_1].quotes.update(G[p_2].quotes)$
        $G[p_1].other\_names.update(G[p_2].other\_names)$
    else
        $G[p_1].parent \leftarrow p_2$
        $G[p_2].rank \leftarrow G[p_2].rank + G[p_1].rank$
        $G[p_2].quotes.update(G[p_1].quotes)$
        $G[p_2].other\_names.update(G[p_1].other\_names)$
```

Algorithm 4: FIND-FEATURES

```
Input: $A[(i_1, aspect_1, sentiment_1, quotes_1), ..., (i_n, aspect_n, sentiment_n, quotes_n)]$
Output: $F$
for each sentiment e in [Positive, Negative] do
    $A_e \leftarrow A[sentiment_i = e]$/* Find elements of A where sentiment is e */
    /* Find unique aspects, their counts, & combine all representative quotes in
       a list */
    $A_c \leftarrow$
      $A_e[(i_1, aspect_1, count_1, quotes_1[q_{11}, ..., q_{1k}]), ..., (i_m, aspect_m, count_m, quotes_m[q_{m1}, ..., q_{ml}])]$

    $G \leftarrow BUILD - GRAPH(A_c)$ /* Call Algorithm 2 */
    for each node_id u in G do
        for each node_id v in G and u != v do
            $UNION(G, u, v)$ /* Call Algorithm 3 */
    /* After the above process, we will be left with the merged nodes, where the
       set of parents indicate the clusters. */
    $F_e \leftarrow [\,]$
    for each parent p in G do
        $N \leftarrow \varnothing$
        $N.name \leftarrow G[p].name$
        $N.rank \leftarrow G[p].rank$
        $N.quotes \leftarrow G[p].quotes$
        $N.other\_names \leftarrow G[p].other\_names$
        $F_e.add(N)$
    $F.add(F_e)$
$return\ F$
```

	Wilcoxon		Mann-Whitney	
Criteria	statistic	p-value	statistic	p-value
Coherence	368.0	0.0526	14958.0	0.0984
Consistency	1122.0	0.4262	14207.0	0.7217
Fluency	832.0	0.1658	14863.0	0.1773
Relevance	472.5	0.189	14566.0	0.2947

Table 8: Significance test on LLM evaluated ratings on summaries generated from distilled content versus all review content in Section 5.3.1. All p-values are greater than the significance level ($\alpha = 0.05$) indicating none of the differences are significant, which implies summaries generated using distilled content are as good as summaries generated using all review content.

Summary prefixes
Customers appreciate
Customers value
Customers highly value
Customers are impressed with
Customers praise
Customers are positive about
Customers admire
Customers frequently mention
Customers commend
Customers are satisfied with
Customers often highlight
Customers consistently note
Customers find value in
Customers enjoy
Customers are enthusiastic about
Customers are pleased with
Customers recognize
Customers express satisfaction with
Customers love
Customers regard
Customers have good things to say about
Customers are delighted by

Table 9: A list of prefixes we ask an LLM to begin a summary with.

~~~
We have a list of customer reviews for a product. Extract at most 5 features from each REVIEW_TEXT.
Features must be relevant to the product attributes or specifications, they must not be representa-
tive of a person, or an animal, avoid naive features like (best, product, good).

Here is the review list, formatted as "PRODUCT_NAME": "", "REVIEW_TEXT": "", "RVW_ID": ""}]:

--------------
<<REVIEW>>
--------------

Output the feature indices, feature names with at most two words, the representative sentences in
the review, and the associated customer sentiments (Positive or Negative only) in a json object.

**ONLY output the following JSON array.  Do not include any other text.**

```json
[
{"RVW_ID": "", "ID": 0, "ASPECT": "", "SENTIMENT": "Positive" or "Negative", "REPR_SENTENCE": ""},
{"RVW_ID": "", "ID": 1, "ASPECT": "", "SENTIMENT": "Positive" or "Negative", "REPR_SENTENCE": ""}
// ...more objects as needed...
]
```
~~~

Figure 4: LLM prompt for aspect-sentiment extraction.

```
You are a helpful assistant and you are tasked with writing a summary from some given information
about a product. We have a list of PROS and CONS of the product, number of times they were mentio-
ned, and a list of representative quotes speaking about the PROS or CONS.

- Write a short and concise summary with no more than four sentences and no less than three sent-
ences on how customers are speaking about different pros and cons.
- Use the statement '#STATEMENT#' to begin the summary.
- Skip reporting how many times a pro/con was mentioned.
- The summary should only highlight pros and cons that are mentioned frequently.
- The summary should use a short name of the product.
- Avoid or rephrase customer mentioned terms that are derogatory, disrespectful, harmful, sexually
explicit, hate speech, or harassment.


The PROS and CONS are listed below:

====================================
PROS_AND_CONS
====================================

{SUMMARY}
```

Figure 5: LLM prompt for review summarization using condensed features from review distillation. #STATEMENT# is replaced with a random prefix from Table 9. PROS_AND_CONS is replaced with a dictionary like object created from the distilled features.

```
You are a helpful assistant and you are tasked with writing a summary from a list of customer
reviews.

- Write a short and concise summary with no more than four sentences and no less than three
sentences on how customers are speaking about different pros and cons.
- Use the statement '#STATEMENT#' to begin the summary.
- Skip reporting how many times a pro/con was mentioned.
- The summary should only highlight pros and cons that are mentioned frequently.
- The summary should use a short name of the product.
- Avoid or rephrase customer mentioned terms that are derogatory, disrespectful, harmful, sexually
explicit, hate speech, or harassment.


The reviews are listed below:

====================================
ALL_REVIEWS
====================================

{SUMMARY}
```

Figure 6: LLM prompt for review summarization using all available reviews for a product. #STATEMENT# is replaced with a random prefix from Table 9. ALL_REVIEWS is replaced with all available review texts for a product.

Product	$\text{Summary}_{\text{full\_context}}$	$\text{Summary}_{\text{distilled\_context}}$
Microsoft Xbox Series X	Customers are impressed with the Xbox Series X's speed and graphics. They rave about the quick load times and the ability to play games at 4K resolution. However, some customers have noted that the console's storage capacity is limited and that the controller hasn't been significantly upgraded.	Customers are impressed with the Xbox Series X's graphics, load times, and Game Pass. The graphics are described as being "amazing" and "impressive," while the load times are "insanely fast" and "almost non-existent." Game Pass is praised for its "massive library of games" and its value for money. However, some customers have expressed concerns about the limited storage space, with some suggesting that users should consider adding external storage.
Apple 10.2-Inch iPad (9th generation)	Customers appreciate the iPad's ease of use, speed, and clear display. They find it perfect for everyday tasks like browsing the web, watching videos, and reading. However, some customers note that the battery life could be better and that the iPad may not be powerful enough for heavy-duty tasks.	Customers appreciate the 10.2-Inch iPad (9th Generation) for its speed, ease of use, and screen size. They find it to be a fast and responsive device that is easy to navigate and has a large enough screen for comfortable viewing. Many customers also highlight the iPad's affordability, making it a great value for the features it offers.
Powerbeats Pro Wireless Earphones	Customers express satisfaction with the Powerbeats Pro's secure fit, especially during workouts. They also praise the sound quality, noting its clarity and powerful bass. However, some customers report connectivity issues, particularly with Android devices, and find the charging case bulky. Additionally, some users find the earbud design uncomfortable for extended wear.	Customers express satisfaction with Powerbeats Pro's sound quality, fit, and battery life. Many appreciate the secure fit, especially during workouts. However, some customers have expressed concerns about the charging process, noting that the earbuds may not charge evenly or connect reliably. Additionally, a few customers have mentioned the high price point as a drawback.
Apple AirPods Max	Customers recognize the Apple AirPods Max as having excellent sound quality and noise cancellation. They appreciate the seamless integration with other Apple devices and the comfortable fit. However, customers also note that the AirPods Max are heavy and expensive, and some find the case to be awkward.	Customers recognize the AirPods Max (Lightning) as having excellent sound quality and being very comfortable. Many customers appreciate the long battery life and the ease of use with other Apple products. However, some customers find the price to be too high and have expressed concerns about the weight of the headphones.
MacBook Air 13.6"	Customers are impressed with the MacBook Air's sleek design, lightweight build, and fast performance. They particularly appreciate the long battery life and the seamless integration with other Apple products. However, some customers have noted that the laptop can be prone to fingerprints and that the base storage option may not be sufficient for everyone.	Customers are impressed with the MacBook Air 13.6" for its speed, battery life, and M2 chip. The laptop is praised for its fast performance, long battery life, and the powerful M2 chip that delivers impressive performance. However, some customers have mentioned that the laptop is prone to fingerprints and that the charging port can be problematic.
Logitech MS Master 3S Wireless Lase Mouse	Customers find value in the MX Master 3S mouse's ergonomic design, which provides comfort during extended use. The mouse's dual scroll wheels, including a horizontal scroll wheel, are highly praised for their functionality and efficiency. However, some customers have noted that the mouse's click buttons feel less premium than other Logitech mice. Additionally, some users have found the mouse's size and shape to be slightly different from previous models, which may not be ideal for all hand sizes.	Customers find value in the MX Master 3S Wireless Laser Mouse's scroll wheel, which they find to be very useful for both work and gaming. They also appreciate the mouse's ergonomic design, which helps to prevent discomfort during long work sessions. Some customers have expressed a desire for a USB-C connector instead of the current USB-A connector.
Epson EcoTank ET-2800	Customers consistently note the Epson EcoTank printer is easy to set up and use, with many praising its wireless capabilities and the convenience of refillable ink tanks. While the printer is generally well-received for its print quality and cost-effectiveness, some users have reported issues with ink refilling and occasional jamming. The printer's small screen and reliance on a mobile app for some functions have also been cited as drawbacks by some customers.	Customers consistently note the EcoTank ET-2800's excellent print quality, with many praising its ability to produce clear, colorful prints. They also appreciate the printer's ease of setup and installation. However, some customers have reported issues with ink refilling, and a few have mentioned that the printer's small screen can make it difficult to operate.

Table 10: A sample list of summaries generated from using all available reviews ($\text{Summary}_{\text{full\_context}}$) for a product versus review distillation content ($\text{Summary}_{\text{distilled\_context}}$).

```
You will be given one summary written for a product.
Your task is to rate the summary on one metric.

Please make sure you read and understand these instructions carefully. Please keep this document
open while reviewing, and refer to it as needed.

Evaluation Criteria:
Coherence (1-5) - the collective quality of all sentences. We align this dimension with the DUC
quality question of structure and coherence whereby ”the summary should be well-structured and
well-organized. The summary should not just be a heap of related information, but should build
from sentence to a coherent body of information about a topic.”

Evaluation Steps:
1. Read the customer reviews about a product carefully and identify the main pros and cons.
2. Read the summary and compare it to the given reviews. Check if the summary covers the main pros
and cons of the product, and if it presents them in a clear and logical order.
3. Assign a score for coherence on a scale of 1 to 5, where 1 is the lowest and 5 is the highest
based on the Evaluation Criteria.

CUSTOMER REVIEWS:
=================================
<<REVIEWS>>
=================================

SUMMARY:
---------------------------------
<<SUMMARY>>
---------------------------------

**Output only a score between 1 to 5**
```

Figure 7: LLM prompt for rating summaries on the evaluation criteria coherence.

```
You will be given one summary written for a product.
Your task is to rate the summary on one metric.

Please make sure you read and understand these instructions carefully. Please keep this document
open while reviewing, and refer to it as needed.

Evaluation Criteria:
Consistency (1-5) -  the factual alignment between the summary and the summarized source. A factually
consistent summary contains only statements that are entailed by the source document.

Evaluation Steps:
1. Read the customer reviews about a product carefully and identify the main pros and cons.
2. Read the summary and compare it to the given reviews. Check if the summary contains any factual
errors that are not supported by the given reviews.
3. Check if the number of sentences in the summary is 3 to 4.
4. Assign a score for consistency on a scale of 1 to 5, where 1 is the lowest and 5 is the highest
based on the Evaluation Criteria.

CUSTOMER REVIEWS:
=================================
<<REVIEWS>>
=================================

SUMMARY:
---------------------------------
<<SUMMARY>>
---------------------------------

**Output only a score between 1 to 5**
```

Figure 8: LLM prompt for rating summaries on the evaluation criteria consistency.

```
You will be given one summary written for a product.
Your task is to rate the summary on one metric.

Please make sure you read and understand these instructions carefully. Please keep this document
open while reviewing, and refer to it as needed.

Evaluation Criteria:
Fluency (1-5) -  the quality of the summary in terms of grammar, spelling, punctuation, word choice,
and sentence structure.

Evaluation Steps:
1. Read the summary carefully.
2. Check if the summary has any errors related to grammar, spelling, and punctuation. Penalize a
summary that has such errors.
3. Asses the word choice and sentence structure. Penalize a summary that has long and complex
sentences.
4. Assign a score for fluency on a scale of 1 to 5, where 1 is the lowest and 5 is the highest
based on the Evaluation Criteria.


CUSTOMER REVIEWS:
=================================
<<REVIEWS>>
=================================

SUMMARY:
---------------------------------
<<SUMMARY>>
---------------------------------

**Output only a score between 1 to 5**
```

Figure 9: LLM prompt for rating summaries on the evaluation criteria fluency.

```
You will be given one summary written for a product.
Your task is to rate the summary on one metric.

Please make sure you read and understand these instructions carefully. Please keep this document
open while reviewing, and refer to it as needed.

Evaluation Criteria:
Relevance (1-5) -  selection of important content from the source. The summary should include only
important information from the customer reviews.

Evaluation Steps:
1. Read the customer reviews about a product carefully and identify the main pros and cons.
2. Read the summary and compare it to the given reviews. Assess how well the summary covers the
main pros and cons from the reviews.
3. If a pro or con is mentioned in only one review it should not be counted as a credible pro/con.
Penalize summaries that contain such cases.
4. Assign a score for relevance on a scale of 1 to 5, where 1 is the lowest and 5 is the highest
based on the Evaluation Criteria.

CUSTOMER REVIEWS:
=================================
<<REVIEWS>>
=================================

SUMMARY:
---------------------------------
<<SUMMARY>>
---------------------------------

**Output only a score between 1 to 5**
```

Figure 10: LLM prompt for rating summaries on the evaluation criteria relevance.