Extracting Root Problems from User Pain Points via a Semantic Feedback Analysis System

Jake Terrill [*1] **Lukas Dendrolivanos** [*1] **Benjamin Corter** [*1] **Zhen Xu** [*1]

Abstract

Innovation, design, and new product development in general, relies heavily on uncovering novel understandings of users. Identifying the root causes of user dissatisfaction is critical in the creation of something truly useful. However, aggregating and analyzing complaints from diverse sources, such as reviews, support tickets, and surveys manually is often labor-intensive, subjective, and difficult to scale. To address this challenge, we present an automated pipeline for extracting root problems and actionable recommendations from user feedback. Our approach leverages a type of Retrieval Augmented Synthesis framework: given a query describing a specific issue, the system employs dense embeddings to retrieve semantically analogous historical pain points. Subsequently, a Large Language Model, using few-shot prompting, synthesizes retrieved examples to identify candidate root causes and provide evidence-backed suggestions.

1. Introduction

Modern enterprises accumulate vast amounts of qualitative feedback on their products and services, ranging from app store reviews and survey responses to support tickets and internal bug reports. While these textual data offer rich insights into user challenges, they are rarely systematically analyzed due to the sheer volume of information. Consequently, critical issues often remain undetected and unresolved for extended periods, buried amidst unstructured data (Kumar et al., 2012).

To address this challenge, this work explores how recent advances in Large Language Models (LLMs) can automate the analysis of user feedback (Vaswani et al., 2017; Brown et al., 2020). Our primary objective is to identify root causes and generate actionable suggestions from user pain points. Specifically, we investigate the underlying factors contributing to specific user problems, the common themes connecting these complaints, and the concrete changes required to mitigate them. We propose a Retrieval-Augmented Synthesis (RAS) framework for synthesizing semantic insights from large-scale unstructured text (Lewis et al., 2020). Unlike manual analysis, which is time-consuming and prone to bias, our approach leverages semantic search to retrieve relevant historical context before synthesizing potential root causes (Karpukhin et al., 2020). This allows for a scalable, data-driven identification of systemic issues that goes beyond simple keyword matching. Our main contributions are summarized as follows:

1. **Curated Pain Point Database.** We construct a high-quality, standardized database of user pain points, rigorously cleaned and extracted from large-scale user reviews, providing a robust foundation for systematic feedback analysis.
2. **Context-Aware RAS Synthesis.** We propose a RAS-based framework that synthesizes insights across similar historical complaints, yielding more accurate and generalizable root-cause identification than direct LLM inference on isolated feedback.
3. **Efficient Retrieval Strategy.** We integrate K-means clustering to pre-classify pain points within the embedding space, significantly reducing the search space and improving retrieval latency and system scalability.

In a qualitative evaluation on Amazon headphone reviews, our system reliably retrieves coherent neighborhoods of complaints and surfaces cross-cutting root problems around reliability, comfort, and communication quality.

2. Related Work

User Feedback Analysis and Root Cause Discovery. The history of User Feedback Analysis (UFA) progressed from early rule-based and statistical methods for text classification (Liu, 2022) to the integration of unsupervised machine learning, where Topic Modeling became the dominant paradigm for scaling the identification of thematic issues (Blei et al., 2003; Chang et al., 2009). However, current engineering practices indicate that Root Cause Analysis (RCA) remains a highly manual, labor-intensive process, requiring expert intervention to interpret aggregated data

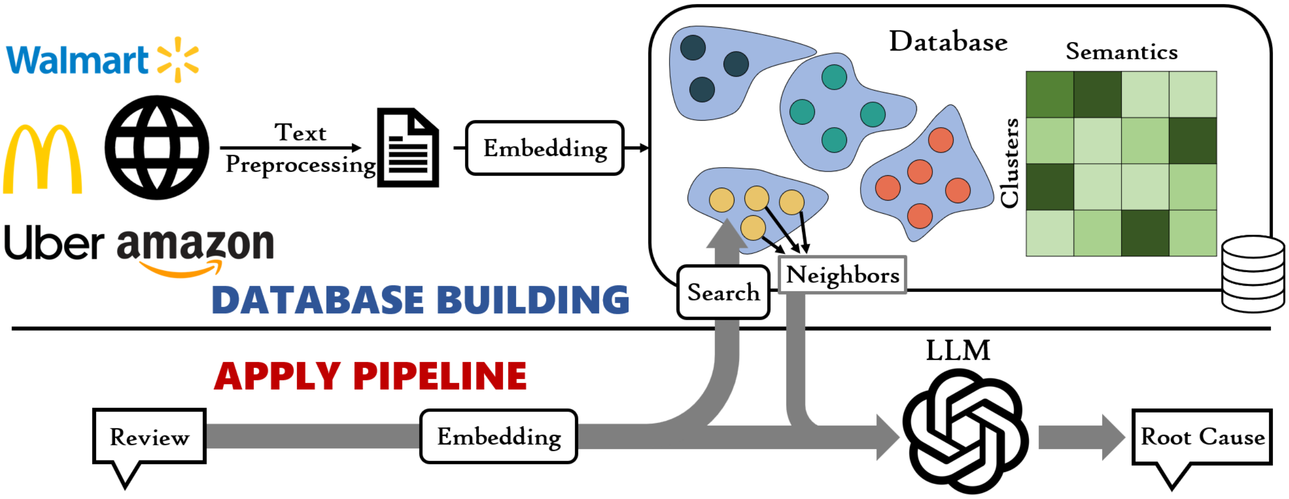

Figure 1. Overview of the RAG pipeline.

and perform the necessary diagnostic inference from themes to underlying problems (Lin et al., 2020).

Semantic Embeddings and Similarity Search. Dense Retrieval, underpinned by neural networks, fundamentally relies on Semantic Text Embeddings (Reimers & Gurevych, 2019) to map textual inputs into a high-dimensional space. This enables powerful Semantic Search (Cer et al., 2018), which has been widely demonstrated to provide more accurate, contextually relevant retrieval than traditional keyword matching. Retrieval from this space typically uses k-Nearest Neighbors (kNN) methods. To ensure efficiency and scalability with large, industrial-scale datasets, the field employs Approximate Nearest Neighbors (ANN) techniques, which serve as the standard backbone of modern knowledge retrieval systems (Johnson et al., 2019).

Retrieval Augmented Generation with LLMs. Large Language Models (LLMs) possess powerful reasoning and synthesis capabilities. Yet, their utility is often constrained by a reliance on parametric knowledge, leading to factual inaccuracies or ”hallucination” when addressing domain-specific queries (Brown et al., 2020; Bubeck et al., 2023). The field of information retrieval has addressed this by developing Retrieval Augmented Generation (RAG), which mitigates these limitations by explicitly conditioning the LLM’s response on external, verifiable knowledge (Lewis et al., 2020). In the RAG paradigm, an external source of documents provides evidence for the model’s hypotheses. This mechanism ensures that the generation process is grounded in concrete, retrieved information (Gao et al., 2023).

3. Methodology

3.1. Retrieval-Augmented Synthesis

Our methodology employs a Retrieval-Augmented Synthesis pipeline designed to identify the Root Cause directly from user Reviews. The pipeline operates in two distinct phases: the Database Building phase (RAS Implement) and the Application Pipeline (Figure 1). In the Database Building phase, Text gathered from various sources, such as reviews from Walmart, McDonald’s, Uber, and Amazon, undergoes preprocessing before being converted into vector Embeddings. These Embeddings are then stored in a central Database. The Application Pipeline initiates when a new Review is processed to obtain its vector Embedding. This Review embedding is used to perform a Database Search and retrieve a set of relevant Neighbors. These Neighbors provide the necessary contextual evidence to the Large Language Model, which synthesizes it to generate the final Root Cause.

3.2. Semantic Clustering

To optimize the retrieval process, we integrate Semantic Clustering into the Database Building stage. This clustering process structures the stored Embeddings based on Semantics to form distinct groups (Clusters) of similar pain points. This structural organization provides two key benefits for the RAS pipeline.

First, by pre-classifying historical feedback into K distinct clusters (where $K \ll N$, N being the total number of data points), the retrieval strategy avoids a brute-force search across the entire dataset. Without clustering, the theoretical search complexity is $O(N \cdot D)$. By contrast, when searching within the clusters, the search is often limited to a subset of $\hat{N}$ points, significantly reducing the effective search complexity and improving retrieval speed and system scalability. The maximum search complexity becomes approximately $O(K \cdot D + \hat{N} \cdot D)$, demonstrating an apparent reduction in the search space.

Second, the explicit grouping of individual embeddings into discernible clusters within the Database enables semantic analysis of each group, enhancing the interpretability and reliability of the overall diagnosis. This optimization ensures that the Search operation retrieves the most contextually relevant Neighbors, thereby strengthening the LLM's ability to synthesize an accurate Root Cause from the provided evidence.

3.3. Clustering Validation and Quality Assessment

K-Means clustering ($k = 25$, `random_state=42`) was applied to the 2,058 review embeddings generated by OpenAI's `text-embedding-3-small` model. The resulting clusters exhibit a balanced size distribution (mean: 82.3 reviews, std: 21.1, range: 51–134) with no orphaned or dominant clusters, suggesting that the partitioning reflects natural semantic structure within the embedding space.

The observed inertia of 995.61 is appropriate for normalized embeddings on a unit hypersphere and corresponds to a 19.51% reduction in variance compared to the unclustered dataset. Principal Component Analysis further validates the embedding quality: the first principal component explains only 7.33% of the variance (indicating no dimensional collapse), while the first 50 components capture 53.57% of the total variance, demonstrating a healthy, high-dimensional representation.

Cluster tightness is similarly consistent. The per-cluster inertia values (mean: 39.82, std: 8.22) show that clusters are comparably compact, which is essential for retrieval quality. In the downstream search stage, limiting retrieval to the top 3–5 most relevant clusters reduces the effective search space from 2,058 reviews to approximately 250–400 reviews without sacrificing semantic relevance. This provides a strong foundation for efficient, cluster-pruned retrieval in the overall RAS pipeline.

We also compared cluster-based retrieval to a brute-force search to see whether clustering changed the quality of the results. The two methods produced extremely similar relevance scores, with an average semantic difference of only 0.0082 (less than a 1% change) showing that clustering does not meaningfully distort which reviews are considered most relevant. The actual overlap in the specific reviews returned was lower than expected (60% instead of the 85%+ we would ideally want), but this is likely due to sampling too few clusters during retrieval; increasing the number of clusters checked from 5 to around 10–15 should close this gap. Even with this conservative setup, cluster-based retrieval was 5.5× faster than brute force. With only 2,000 reviews in the dataset, this speed gap is already notable and will grow significantly as the database expands, highlighting clustering as an efficient retrieval strategy.

4. Results

We qualitatively evaluated our proposed RAS pipeline using 15 representative queries focused on the earphones/earbuds/headphones domain, based on Amazon reviews, and identified three with the highest cosine similarity scores: Battery and Usage Patterns, Comfort and Wearability, and Call Quality and Communication. For each query, the system retrieves the top-k nearest neighbor reviews from the clustered embedding space and then synthesizes three root problems from this neighborhood using the LLM.

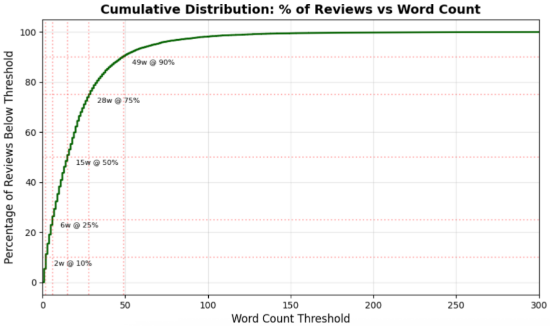

Figure 2. Cumulative review length distribution by word count.

4.1. Dataset

The selected dataset is a curated collection of consumer reviews, focusing explicitly on the earphones, earbuds, and headphones subcategory within Amazon's electronics vertical. This coherent domain, characterized by similar vocabulary and meaningful product differentiation, provides a clear semantic space for developing a professional Machine Learning text pipeline. The data was retrieved from Kaggle and exported to a CSV file, with each record containing the product ID, product name, review text, and corresponding rating. To ensure data quality, we applied two primary filters. **(1) Review Length.** We reduced the dataset to include only reviews with 15 or more words. This threshold was chosen to filter out superficial content (e.g., ”Good product,” ”Bad Price”) and retain reviews that demonstrated depth and reasoning. Visualization (Figure 2) confirmed that the 15-word threshold captured 50% of the available data while ensuring high quality. **(2) Product Balancing.** To prevent any single product from disproportionately influencing the cumulative dataset (and subsequently the product representations), we capped the number of reviews per product at a maximum of 400. This value was selected based on the median review count across all ten products, ensuring a high-quality, balanced representation of multiple products and semantic spaces.

4.2. Retrieval Behavior Across Queries

Across all three queries, the system consistently retrieves a compact, semantically coherent set of 20 reviews from a small number of products, indicating that the clustered embedding space can localize related complaints. For Battery & Usage Patterns, the system returns 20 reviews from 6 products (dominated by JBL T110BT, boAt Rockerz 255, and JBL T205BT), with similarity scores ranging from 0.501 to 0.580 (avg. 0.527) and an average rating of 3.4/5. For Comfort & Wearability, it retrieves 20 reviews from 5 products, heavily concentrated on JBL T205BT (14/20), with similarity scores between 0.524 and 0.612 (avg. 0.554) and an average rating of 4.0/5. Finally, for Call Quality & Communication, it retrieves 20 reviews from 5 products (10 from JBL T110BT), with similarity scores ranging from 0.530 to 0.609 (avg. 0.551) and an average rating of 3.0/5. In all three cases, the retrieved neighborhoods are semantically coherent around the target issue (such as battery unreliability, comfort and fit, or microphone performance) rather than general product sentiment.

4.3. Synthesized Root Problems and Diagnostic Utility

Conditioned on these neighborhoods, the LLM produces three root problems per query. **Regarding Battery & Usage Patterns**, the analysis identifies a need for reliable battery performance (no unexpected shutdowns, no rapid drain, no frequent recharging), a desire for clear battery communication (no missing or misleading low-battery indicators), and an expectation of an effortless usage experience, avoiding frequent disconnections and awkward controls. For **Comfort & Wearability**, the system highlights the need for comfort during extended use (severe ear pain after 30–120 minutes), a secure fit during active use, and the expectation of customization for individual fit in terms of tip sizes. Lastly, for **Call Quality & Communication**, the results point to a need for clear communication during calls (due to background noise and muffled voice pickup), an expectation of durable microphone performance, and a desire for an effortless, hands-free experience.

Taken together, these three cases illustrate that the proposed pipeline does more than summarize complaints; it consistently transforms different, product-specific reviews into a small set of interpretable, higher-level design and requirement statements. Across domains, users express recurring expectations for reliability over time, transparency of system feedback, physical comfort and fit, and low-friction everyday use. The generated root problems closely resemble the kinds of abstractions produced by manual root cause analysis, suggesting that clustered retrieval coupled with LLM-based synthesis can provide actionable diagnostic insight from large-scale user feedback.

4.4. Comparison with Stronger Model Outputs

To evaluate whether larger models yield qualitatively different diagnostic structure, we additionally generated root problems using `gpt-5-pro-2025-10-06` while holding the remainder of the pipeline fixed. The higher-capacity model produced results that were broadly aligned with those from the smaller model conditioned synthesis, but demonstrated noticeably stronger hierarchical organization and clearer articulation of latent user needs.

For *Battery & Usage Patterns*, GPT-5-Pro separated issues into (1) endurance that matches real-life multi-hour workflows and (2) predictable low-battery behavior with reliable recovery, each supported by explicit evidence from retrieved reviews. For *Comfort & Wearability*, the model distinguished between (1) long-duration, pressure-free fit across diverse ear anatomies and (2) stable wear during movement without over-tightening—mirroring themes extracted by the smaller model, but stated in more generalizable design language. Similarly, for *Call Quality & Communication*, GPT-5-Pro identified (1) robustness to background noise and (2) low-friction, dependable call performance, capturing the same failure modes as before but with clearer abstraction of the underlying root needs.

Overall, GPT-5-Pro did not alter the semantic conclusions of the pipeline; instead, it provided more sharply defined, evidence-grounded formulations of the same latent problems. This suggests that, while our RAS pipeline is effective even with lightweight models, higher-capacity LLMs can enhance the clarity and generality of the synthesized root problems without changing their fundamental structure. This showcases the scalability that this pipeline embodies, if it were to be taken to a next-level application.

5. Discussion

The presented RAS-based methodology offers a structured, automated solution for diagnostic inference, effectively addressing the traditional bottleneck of manual Root Cause Analysis (RCA). This automated structure is designed to yield high-quality synthesis by grounding the Large Language Model in concrete evidence retrieved from the database. Notably, this high performance is achieved efficiently by integrating Semantic Clustering into the Database Building stage. This integration validates the architectural choice, ensuring that performance gains are achieved without the computationally prohibitive brute-force retrieval. Furthermore, the explicit clustering of historical evidence enhances the interpretability and reliability of the diagnosis by making the LLM's output traceable to semantically coherent groups. While the current framework is robust, future work should explore optimizing the clustered space using more advanced Approximate Nearest Neighbors (ANN) techniques and developing continuous learning mechanisms to adapt the database to evolving user feedback patterns dynamically.

References

Blei, D. M., Ng, A. Y., and Jordan, M. I. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan): 993–1022, 2003.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., Lee, P., Lee, Y. T., Li, Y., Lundberg, S., et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*, 2023.

Cer, D., Yang, Y., Kong, S.-y., Hua, N., Limtiaco, N., John, R. S., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C., et al. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*, 2018.

Chang, J., Gerrish, S., Wang, C., Boyd-Graber, J., and Blei, D. Reading tea leaves: How humans interpret topic models. *Advances in neural information processing systems*, 22, 2009.

Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, H., and Wang, H. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2(1), 2023.

Johnson, J., Douze, M., and Jégou, H. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7 (3):535–547, 2019.

Karpukhin, V., Oguz, B., Min, S., Lewis, P. S., Wu, L., Edunov, S., Chen, D., and Yih, W.-t. Dense passage retrieval for open-domain question answering. In *EMNLP (1)*, pp. 6769–6781, 2020.

Kumar, A., Sebastian, T. M., et al. Sentiment analysis: A perspective on its past, present and future. *International Journal of Intelligent Systems and Applications*, 4(10): 1–14, 2012.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474, 2020.

Lin, F., Muzumdar, K., Laptev, N. P., Curelea, M.-V., Lee, S., and Sankar, S. Fast dimensional analysis for root cause investigation in a large-scale service environment. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 4(2):1–23, 2020.

Liu, B. *Sentiment analysis and opinion mining*. Springer Nature, 2022.

Reimers, N. and Gurevych, I. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.