A Hybrid Supervised-LLM Pipeline for Actionable Suggestion Mining in Unstructured Customer Reviews

Aakash Trivedi[1] **Aniket Upadhyay**[1] **Pratik Narang**[1] **Dhruv Kumar**[1]
Praveen Kumar[2]
[1]Department of Computer Science & Information Systems,
Birla Institute of Technology and Science, Pilani, India
f20191076P@alumni.bits-pilani.ac.in
{p20241007, pratik.narang, dhruv.kumar}@pilani.bits-pilani.ac.in
[2]Birdeye Inc., Palo Alto, California, USA
praveen.kumar1@birdeye.com

Abstract

Extracting actionable suggestions from customer reviews is essential for operational decision-making, yet these directives are often embedded within mixed-intent, unstructured text. Existing approaches either classify suggestion-bearing sentences or generate high-level summaries, but rarely isolate the precise improvement instructions businesses need. We evaluate a hybrid pipeline combining a high-recall RoBERTa classifier trained with a precision–recall surrogate to reduce unrecoverable false negatives with a controlled, instruction-tuned LLM for suggestion extraction, categorization, clustering, and summarization. Across real-world hospitality and food datasets, the hybrid system outperforms prompt-only, rule-based, and classifier-only baselines in extraction accuracy and cluster coherence. Human evaluations further confirm that the resulting suggestions and summaries are clear, faithful, and interpretable. Overall, our results show that hybrid reasoning architectures achieve meaningful improvements fine-grained actionable suggestion mining while highlighting challenges in domain adaptation and efficient local deployment.

1 Introduction

Customer reviews contain valuable signals for service improvement, but explicit suggestions, concrete requests for what should be fixed, added, or improved are typically rare and embedded within long, mixed-intent narratives. In this work, we define an actionable suggestion as an explicit, business-directed suggestion that specifies a concrete operational change (e.g., “Add more vegetarian options”), rather than general opinions, complaints, or advice to other customers. Automatically identifying these actionable spans remains challenging, reviews blend praise, complaints, stories, and user-to-user advice, making heuristic or manual approaches unreliable at scale.

Prior work on suggestion mining has focused largely on sentence-level detection (Negi and Buitelaar, 2015; Wicaksono and Myaeng, 2013; Dong et al., 2017), which identifies the presence of a suggestion but does not extract the actionable phrase, handle multi-sentence directives, or distinguish business-directed improvements from general opinions. Transformer-based and domain-adaptive models (Joshi et al., 2020; Riaz et al., 2024) improve detection but still frame the task as classification rather than full extraction.

Related research in opinion summarization and theme modeling (Angelidis and Lapata, 2021; Mukku and Mukku, 2024; Nayeem and Rafiei, 2024) captures high-level topics, but does not surface the specific improvements needed for operational decision-making. Meanwhile, LLMs offer strong structured extraction capabilities (Ouyang et al., 2022), yet LLM-only methods suffer from hallucination (Ji et al., 2023), inconsistent span boundaries (Koto et al., 2022), and low recall for infrequent suggestion types. Conversely, rule-based or classifier-only systems are brittle and lack generalization.

We investigate whether a hybrid architecture, pairing a high-recall supervised classifier with controlled LLM-based extraction, categorization, clustering, and summarization can more reliably surface actionable suggestions from reviews. We frame this as end-to-end *actionability extraction*, detecting suggestion-bearing reviews, isolating explicit improvement directives, grouping them semantically, and producing concise summaries suitable for decision-making.

Our contributions are:

- A recall-oriented RoBERTa classifier trained with a precision–recall surrogate objective to reduce unrecoverable false negatives, while maintaining comparable precision.
- An instruction-tuned, quantized LLM for con-

Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 5: Industry Track), pages 823–836
March 25-27, 2026 ©2026 Association for Computational Linguistics

trolled extraction, categorization, clustering, and summarization.

- Extensive comparisons against prompt-only LLMs, rule-based systems, classifier-only pipelines, and end-to-end LLM methods.

- Comprehensive evaluation of extraction, category assignment, clustering, and summarization using automatic metrics, human judgments, and ablations.

By focusing on explicit, operationally meaningful suggestions rather than generic opinions, we show that a hybrid approach mitigates the weaknesses of classifier-only and LLM-only systems, offering an approach suitable for large-scale operational settings.

2 Related Work

2.1 Suggestion Mining

Early work framed suggestion mining as binary classification, using benchmarks by Negi and Buitelaar (2015) and linguistic-pattern methods (Wicaksono and Myaeng, 2013). Neural models with attention (Dong et al., 2017) and transformer variants such as TransLSTM (Riaz et al., 2024) improved detection, while span-based architectures (e.g., SpanBERT; Joshi et al., 2020) support finer extraction. However, these systems largely detect suggestion presence rather than extracting explicit actionable spans or handling multi-sentence suggestions.

2.2 Opinion Summarization and Theme Modeling

Opinion summarization condenses reviews into themes or aspect-level insights. Topic models (Blei et al., 2003; Dieng et al., 2020) and modern abstractive systems (Bražinskas et al., 2020; Angelidis and Lapata, 2021) produce high-level representations, and domain-specific models such as InsightNet (Mukku and Mukku, 2024) and LFOSum (Nayeem and Rafiei, 2024) cluster user opinions. Yet these approaches emphasize broad aspects rather than the precise improvements customers request, limiting actionability.

2.3 Hybrid Approaches and Multi-Stage Reasoning

Hybrid pipelines combining targeted classifiers with downstream reasoning are common in fact verification (Thorne et al., 2018), relation extraction (Zhou and Xu, 2018), and retrieval-augmented QA (Chen et al., 2017). Surveys highlight classifier-driven constraints as a method to reduce LLM hallucination (Wu et al., 2023). However, such hybridization has not been explored for actionable suggestion extraction nor evaluated across downstream stages (clustering, summarization).

2.4 LLMs for Structured Extraction

Instruction-tuned LLMs, including GPT models (Ouyang et al., 2022), LLaMA (Touvron et al., 2023), Mistral (Jiang et al., 2023), and Gemma (Team, 2024) enable strong structured extraction, yet LLM-only pipelines remain prone to hallucination (Ji et al., 2023), unstable span boundaries (Koto et al., 2022), and degraded performance on large input batches. Our approach mitigates these issues using classifier gating and tightly controlled prompting in a multi-stage pipeline.

2.5 Positioning

Where prior work targets suggestion detection, theme discovery, or high-level summarization, we focus on extracting *explicit, actionable* suggestions and organizing them into interpretable structures. Our evaluation spans classification, extraction, categorization, clustering, summarization, cross-domain generalization, and ablations, providing the comprehensive study of end-to-end actionability extraction.

3 Methodology

This section describes the design of our hybrid suggestion-mining pipeline. We first provide a high-level system overview (Section 3.1), followed by the classifier training procedure (Section 3.2), the LLM-based components (Section 3.3), and the prioritization logic used in downstream applications (Section 3.4).

3.1 System Overview

The proposed system converts raw customer reviews into structured, actionable suggestions through a multi-stage hybrid pipeline. A fine-tuned RoBERTa classifier (Liu et al., 2019) performs binary classification to identify reviews that contain at least one explicit, business-directed actionable suggestion, ensuring that only relevant inputs propagate downstream. Subsequent stages, suggestion extraction, category assignment, clustering, and summarization are performed by an instruction-tuned and quantized Ollama Gemma-3 model. Figure 1 illustrates the overall workflow. Appendix K illustrates the execution of the pipeline on real-world review examples.

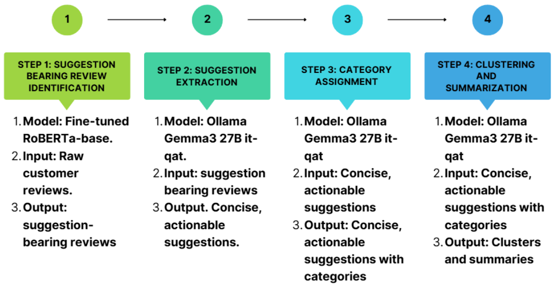
1
2
3
4
STEP 1: SUGGESTION BEARING REVIEW IDENTIFICATION
1. Model: Fine-tuned RoBERTa-base.
2. Input: Raw customer reviews.
3. Output: suggestion-bearing reviews
STEP 2: SUGGESTION EXTRACTION
1. Model: Ollama Gemma3 27B it-qat.
2. Input: suggestion bearing reviews
3. Output. Concise, actionable suggestions.
STEP 3: CATEGORY ASSIGNMENT
1. Model: Ollama Gemma3 27B it-qat
2. Input: Concise, actionable suggestions
3. Output: Concise, actionable suggestions with categories
STEP 4: CLUSTERING AND SUMMARIZATION
1. Model: Ollama Gemma3 27B it-qat
2. Input: Concise, actionable suggestions with categories
3. Output: Clusters and summaries

Figure 1: Overall process flow of the proposed method.

The design of this pipeline, notably the separation of classification and LLM-based reasoning is motivated by the need for high recall in the first stage (failing to detect a suggestion is unrecoverable) and the strong abstraction and rewriting capabilities of LLMs in the subsequent stages.

3.2 Classifier Training and Optimization

3.2.1 Dataset and Model Choice

We trained the classifier on a proprietary dataset of 1,110 reviews (440 positive, 670 negative). RoBERTa-base was selected after experimentation with multiple models (see Appendix A) due to its favorable trade-off between accuracy and computational cost. A learning-curve analysis (Figure 2) further shows that the classifier saturates at roughly 70% of the training data, indicating that the dataset is sufficiently large for this task.

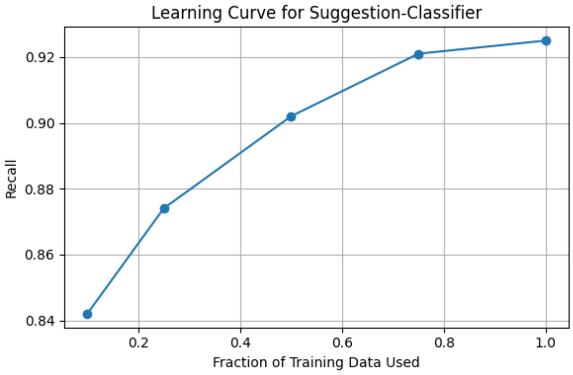

Figure 2: Learning curve showing recall as a function of training data size. Performance saturates around 70% of the dataset, indicating that the dataset is sufficiently large for the classification task.

3.2.2 Hybrid Precision–Recall-Oriented Objective

To encourage high recall while retaining calibrated probabilities, we optimize a hybrid loss combining standard cross-entropy and a differentiable surrogate approximation of the precision–recall curve. For an input x_i with label $y_i \in \{0, 1\}$ and predicted probability p_i:

$$L_{\text{CE}} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log p_i + (1 - y_i)\log(1 - p_i)\right]. \tag{1}$$

Let $s_i = p_i$ denote the predicted score, and let $\{t_k\}_{k=1}^{K}$ be uniformly spaced thresholds in $[0, 1]$. Using the sigmoid function $\sigma(z) = \frac{1}{1+\exp(-z)}$ and a temperature parameter $\tau > 0$, the soft counts of predicted positives (PP) and true positives (TP) at threshold t_k are:

$$\widehat{\text{PP}}(t_k) = \sum_{i=1}^{N} \sigma\left(\frac{s_i - t_k}{\tau}\right), \tag{2}$$

$$\widehat{\text{TP}}(t_k) = \sum_{i=1}^{N} y_i\, \sigma\left(\frac{s_i - t_k}{\tau}\right). \tag{3}$$

The soft precision at threshold t_k is then:

$$\widehat{\text{Precision}}(t_k) = \frac{\widehat{\text{TP}}(t_k)}{\widehat{\text{PP}}(t_k) + \varepsilon}, \tag{4}$$

with ε added for numerical stability. The precision–recall surrogate loss is defined as:

$$L_{\text{PR}} = 1 - \frac{1}{K}\sum_{k=1}^{K}\widehat{\text{Precision}}(t_k). \tag{5}$$

The total training objective is:

$$L_{\text{total}} = \alpha L_{\text{CE}} + (1 - \alpha)\lambda L_{\text{PR}}, \tag{6}$$

where α balances probability learning and λ scales the recall-oriented regularization.

Implementation Details. Complete hyperparameter configuration appears in Appendix B.

3.3 LLM-Based Extraction, Categorization, Clustering and Summarization

We employ the instruction-tuned and quantized Ollama Gemma-3 27B model for suggestion extraction, categorization, clustering, and summarization. Few-shot prompting (Brown et al., 2020) and task-specific prompt templates guide the model to:

1. isolate explicit suggestions from each review,
2. rewrite them into concise, context-complete statements,
3. assign each suggestion to a canonical category,
4. cluster semantically similar suggestions within each category by having the LLM jointly compare all suggestions in that category, identify groups of high-level thematic similarity, and dynamically determine the appropriate number of clusters,
5. produce short, coherent summaries for each cluster.

The LLM operates solely on raw review text and extracted suggestions and does not have access to any human annotations or gold spans. Annotated data is used only for training and evaluating the classifier. These steps enable structured, interpretable grouping of customer feedback while preserving essential semantic distinctions.

Model Selection Process. We evaluated several instruction-tuned LLMs from the HuggingFace and LMArena leaderboards, prioritizing extraction reliability, semantic stability for clustering, and feasibility of local inference. Smaller and mid-scale models (e.g., Qwen2.5-0.5B, Qwen1.5, Mistral-7B) showed inconsistent extraction and unstable semantic groupings (Appendix C). Ollama Gemma-3-27B (quantized) was ultimately selected due to its large context window, high extraction fidelity, and stable, coherent cluster representations, while also producing concise, compact summaries. Full configuration details and prompt templates appear in Appendices I and D.

3.4 Prioritization and Standalone Suggestions

Suggestions that do not fit into any cluster are treated as standalone outputs. During industrial deployment, standalone outputs, while still valuable, are treated as lower-priority insights because clustered suggestions represent feedback raised by multiple customers, indicating greater frequency and operational significance.

4 Experimentation and Results

Our experiments evaluate three research questions:

RQ1 Does the proposed classifier outperform lexical, rule-based, and LLM-only approaches in detecting suggestion-bearing reviews?

RQ2 Does the precision–recall surrogate objective improve recall without sacrificing precision?

RQ3 Does the full hybrid pipeline (classifier + LLM extraction + clustering + summarization) outperform alternative end-to-end baselines in extraction quality, cluster coherence, interpretability, and stability?

All evaluations use held-out datasets from the restaurant and ice-cream domains unless otherwise specified. All experiments were run on our local workstation Appendix J, except the non-quantized Gemma-3 model, which was executed on a separate high-memory machine.

4.1 Dataset Statistics

Actionable suggestions are sparse (13–18%), and review lengths vary widely. Full dataset details and statistics are provided in Appendix E.

4.2 RQ1: Classifier-Level Evaluation

Baselines. To contextualize the performance of the RoBERTa-base classifier, we compare against:

1. **Lexical baseline**: surface-pattern heuristics.
2. **Prompt-only LLM**: Gemma-3 directly classifies raw reviews.
3. **Rule-based**: keyword + dependency templates.

The lexical baseline achieves low recall (0.52) and low precision (0.48). It frequently misclassifies descriptive narratives as suggestions while missing paraphrased directives, leading to a high false-positive rate that makes it unsuitable for downstream extraction and clustering.

The prompt-only LLM obtains higher performance (precision = 0.72, recall = 0.68), but it suffers from two limitations: (i) it often labels implied or indirect opinions as explicit suggestions, reducing precision, and (ii) it is computationally expensive, requiring 3–6 seconds per review (10–15 seconds for long reviews), which makes large-scale deployment infeasible.

The rule-based method achieves moderate precision but very low recall (precision = 0.58, recall = 0.30). Although rule triggers are designed to match explicit imperative constructions, they often fire on spurious cases such as polite suggestions, conditional phrasing, or dependency patterns that match syntactically but lack true directive meaning. These template-level false positives reduce precision relative to the prompt-only LLM, which benefits from stronger contextual reasoning and filters out many superficially similar but non-actionable constructions.

RoBERTa Performance. Table 8 in Appendix F shows that RoBERTa-base achieves strong precision (0.9039) and the best recall (0.9221).

4.3 Cross-Domain Generalization

We further tested the classifier on four additional industries to assess robustness. Recall remained high across domains, though precision varied. See Appendix G for full results.

4.4 RQ2: Effectiveness of the Precision–Recall Surrogate Objective

To isolate the effect of the recall-oriented hybrid loss, we trained the classifier using standard cross-entropy alone. Removing the PR surrogate reduces recall to 0.8873 (–3.49%) with negligible change in precision. Although the gain appears small, even a few recall points correspond to many additional suggestions in large-scale operational settings, and missed items are unrecoverable downstream. Bootstrap testing confirms that the improvement is statistically significant ($p < 0.01$).

4.5 RQ3: End-to-End Pipeline Evaluation

We now evaluate the full pipeline including extraction, categorization, clustering, and summarization against three end-to-end baselines:

- **Prompt-only LLM**: Gemma-3 performs extraction and rewriting without a classifier.
- **Classifier-only pipeline**: classifier + rule-based extraction + clustering.
- **Rule-based end-to-end**: rule-based detection + extraction + clustering.

4.6 Extraction Quality Evaluation

We evaluate extraction quality using 150 manually annotated reviews from both domains. The hybrid pipeline produces *rewritten, canonicalized suggestions* rather than raw spans. These rewrites are necessary for stable downstream category assignment and clustering, as they normalize phrasing and remove irrelevant or fragmented tokens. Consequently, span-matching metrics (Exact/Fuzzy F1) primarily measure lexical overlap and therefore do *not* reflect the extraction objective of our system. We treat semantic metrics (BERTScore, BLEURT) as the primary indicators of correctness, and report span-based scores only for baselines that copy substrings.

We compare four systems: (1) the hybrid pipeline, (2) a prompt-only LLM extractor, (3) a rule-based span extractor, and (4) a T5-base span model.

Model	BERTScore	BLEURT	Exact F1	Fuzzy F1
Hybrid pipeline	0.92	0.89	0.32	0.68
Prompt-only LLM	0.87	0.84	0.56	0.70
T5-base (span)	0.78	0.76	0.72	0.73
Rule-based	0.46	0.44	0.42	0.45

Table 1: Extraction quality. Hybrid outputs are rewritten, thus exact span metrics are not directly comparable.

Discussion. As illustrated in Table 1 the hybrid system achieves the strongest *semantic* extraction quality (BERTScore 0.92, BLEURT 0.89). Its low Exact F1 is expected because rewritten outputs differ lexically from gold spans, whereas span-based models (e.g., T5-base) receive inflated scores by copying text verbatim. Table 2 shows that the hybrid outputs remain semantically faithful even when wording differs.

Cluster Coherence. Cluster quality is measured using Adjusted Mutual Information (AMI), computed with the standard scikit-learn implementation. The hybrid pipeline achieves the highest coherence (AMI = 0.67), surpassing both the prompt-only LLM (0.49) and the classifier-only baseline (0.38).

All systems use the same LLM-based clustering procedure, differences arise solely from the quality of their inputs. The hybrid pipeline provides clean, consistently rewritten suggestions, which reduces lexical variability and enables more stable grouping. Prompt-only extraction produces noisier and occasionally implied suggestions, while classifier-only spans are short and incomplete, leading to fragmented clusters.

We do not report SBERT+HDBSCAN baselines because embedding-based clustering relies on vector similarity rather than the operational themes required for actionable suggestion mining. In preliminary experiments, such methods either over-fragmented paraphrases or over-merged distinct is-

Review Excerpt	Gold Span	Hybrid Output	Notes
“Please add more outdoor seating; it gets crowded.”	add more outdoor seating	Add additional outdoor seating to handle evening crowds.	Meaning preserved; context improved.
“I wish there were more chargers at tables.”	more chargers available at tables	Provide additional table chargers for customers.	Rewritten but semantically equivalent.
“Check-in is slow—please streamline it.”	please streamline it	Streamline the check-in process to reduce delays.	Clarifies the target action.

Table 2: Hybrid outputs are semantically correct even when not span-identical.

sues, producing clusters that were less interpretable for downstream business use. Our evaluation therefore focuses on the LLM-driven clustering mechanism employed by all systems.

Category Assignment Evaluation. We evaluated category assignment on a 150-instance held-out set. The hybrid pipeline achieves the highest accuracy (0.90), followed by the prompt-only LLM (0.78) and the rule-based spans (0.62).

Summarization Evaluation. We evaluate cluster summaries using ROUGE-L and BERTScore (F1).

Model	ROUGE-L	BERTScore (F1)
Hybrid pipeline	0.46	0.91
Prompt-only LLM	0.34	0.86
Rule-based	0.22	0.75

Table 3: Summarization performance for cluster-level summaries.

The hybrid pipeline produces contextually richer, rephrased summaries that differ in wording from the reference, as a result, ROUGE scores are lower, while BERTScore captures semantic similarity and remains high.

Human Evaluation. Three industry experts rated extraction, categorization, clustering, and summarization on a 1–5 Likert scale, with substantial to near-perfect agreement ($\kappa = 0.74$–0.85). Full annotation details are provided in Appendix H. The hybrid pipeline scored highly across all dimensions: extraction (4.0–5.0), categorization (4.0–4.6), clustering (5.0), and summarization (4.6–5.0), indicating strong interpretability and overall pipeline stability.

4.7 Ablation Studies

To quantify the contribution of individual components, we evaluate the pipeline with specific modules removed:

- **No clustering**: interpretability drops by 22% (human-rated).
- **No quantization**: memory usage increases by 2.4× and latency by 47%, with negligible quality change (ΔF1 < 0.01).
- **No PR-loss**: recall drops by 3.49%.
- **No category assignment**: AMI decreases by 0.12.

These ablations show categorization and clustering are essential for coherent downstream insights, while quantization improves deployability with minimal quality loss.

4.8 Error Analysis

Classifier errors mainly stem from sarcastic phrasing and domain-specific terminology that mimics suggestion language. LLM errors are rare but include occasional mis-clustering of closely related suggestions and summaries that could be more concise. These issues point to future improvements in domain-adaptive fine-tuning and prompt refinement.

5 Conclusion

We investigated a hybrid pipeline that combines supervised suggestion detection with LLM-based extraction and structuring. Across extraction accuracy, clustering coherence, and human-rated interpretability, the approach shows consistent gains over prompt-only LLMs, rule-based extractors, and classifier-only variants. The precision–recall surrogate improves recall, which is critical because missed suggestions cannot be recovered. Cross-domain tests show robust recall across real estate, healthcare, finance, and automotive reviews, with some precision loss in domains with specialized terminology. Ablations indicate that clustering and category assignment enhance interpretability, and that quantization improves deployability with minimal quality loss. Remaining challenges include domain-specific phrasing and occasional LLM misclustering. Beyond controlled experiments, the framework has also been applied in a real business context, demonstrating its viability in large-scale operational settings and surfacing practical deployment considerations. Overall, hybrid reasoning pipelines offer a viable strategy for high-recall detection and structured suggestion extraction, with future work in domain-adaptive tuning, multilingual extension, and improved prompt robustness.

6 Limitations

Our study has a few limitations. The use of proprietary review data restricts full reproducibility, as we cannot release the raw text due to confidentiality constraints. While the pipeline maintains strong recall on datasets from unrelated industries such as automotive services, healthcare, and retail banking, its precision varies across domains. Achieving production-level accuracy in these settings will require domain-specific adaptation, since differences in vocabulary, feedback style, and how customers articulate suggestions affect both the classifier and the extraction prompts. Another limitation concerns the clustering stage: although the LLM-based grouping is generally coherent, it can occasionally misassign suggestions to closely related but distinct themes, especially when operational issues share overlapping terminology. These behaviors reflect the sensitivity of the clustering prompts, where minor phrasing changes can shift how the model interprets semantic boundaries. More robust prompt design or lightweight prompt tuning is therefore needed to improve cluster discriminability and reduce cross-topic bleed-over. While raw data cannot be released due to confidentiality constraints, we provide full prompt templates, model configurations, hyperparameters, and hardware specifications to enable faithful reproduction of the pipeline on alternative datasets.

References

Stefanos Angelidis and Mirella Lapata. 2021. Summarizing opinions with gsum. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Algirdas Bražinskas, Róbert Busa-Fekete, and Daniel Preotiuc-Pietro. 2020. Learning to summarize product reviews by exploiting aspect-level ground truth. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 1877–1901.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Adji B. Dieng, Chong Wang, Jianfeng Gao, and John Paisley. 2020. Topic modeling in embedding spaces. In *Transactions of the Association for Computational Linguistics*, volume 8, pages 439–453.

Li Dong, Qi Qian, and Lei Jiang. 2017. Attention-based neural networks for suggestion mining. In *Proceedings of the 2017 Conference on Natural Language Processing (ACL)*.

Zhenzhong Ji, Richard Lee, Joseph Fries, and Roger Levy. 2023. A survey of hallucination in large language models. *ACM Computing Surveys*, 56(12):1–42.

Xinyu Jiang, Min Chen, and Zhen Li. 2023. Mistral: Open-weight instruction-tuned language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Mandar Joshi, Danqi Chen, and Yinhan Liu. 2020. Spanbert: Improving pre-training by representing and predicting spans. In *Transactions of the Association for Computational Linguistics*, volume 8, pages 64–77.

Ryota Koto, Keisuke Yoshida, and Takuya Tanaka. 2022. Span-level inconsistencies in llm-based extraction. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Sai Mukku and Praneeth Mukku. 2024. Insightnet: A neural network for semantic theme extraction. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Mir Tafseer Nayeem and Davood Rafiei. 2024. LFOSum: Large-scale summarization of fine-grained opinions. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Sapna Negi and Paul Buitelaar. 2015. Towards the extraction of customer-to-customer suggestions from reviews. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2159–2167.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Journal of Machine Learning Research*, 23:1–60.

Muhammad Riaz, Ayesha Raza, and Hira Javed. 2024. TransLSTM: Transformer-enhanced LSTM for suggestion mining. *Natural Language Engineering*, 30(2):123–145.

Gemma Team. 2024. Gemma: Instruction-tuned large language model for structured extraction. https://gemma-model.org.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. Fever: a large-scale dataset for fact extraction and verification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Hugo Touvron, Thibaut Martin, Pete Stone, Eric Albert, Amjad Almahairi, Thomas Rault, Victor Dognin, Herman Lespiau, and Sylvain Gelly. 2023. Llama: Open and efficient foundation language models. In *Advances in Neural Information Processing Systems*.

Budi Wicaksono and Sung-Hyon Myaeng. 2013. Mining product improvement suggestions from customer reviews. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL)*.

Yuxin Wu, Lili Wang, and Ming Chen. 2023. A survey of hybrid approaches for nlp tasks: Classifier-language model pipelines. *ACM Computing Surveys*, 56(7):1–36.

Peng Zhou and Wei Xu. 2018. Joint extraction of entities and relations based on a novel tagging scheme. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

A Model Selection Process

Table 4 summarizes the performance of multiple classifier models trained on the same dataset using identical training procedures. The table allows a direct comparison of different model architectures and hyperparameter configurations under consistent training conditions. Based on these results, we identified the top-performing models and further filtered them to select the best candidate for the suggestion extraction task.

Model	Precision	Recall
GPT-2 Small	0.6765	0.7419
GPT-2 Medium	0.9565	0.7097
ROBERTa-Large	0.9615	0.8065
DeBERTa-Large	0.8485	0.9032
BERT-Large	0.8214	0.7419
XLNet-Large	0.8750	0.9032
BART-Large	0.9000	0.8710
ROBERTa-Base	**0.9039**	**0.9221**

Table 4: Performance of various models on the testing dataset.

B Classifier Training Hyperparameters

Table 5 summarizes the full configuration used to train the high-recall RoBERTa-base classifier with the precision–recall surrogate objective.

Parameter	Value
Number of thresholds (K)	25
Temperature (τ)	0.02
Stability constant (ε)	1×10^{-8}
Batch size	16
Optimizer	AdamW
Learning rate	1×10^{-5}
Weight decay	0.01
Warmup ratio	0.1
Loss weight α	0.6
Loss weight λ	1.3
Random seed	888

Table 5: Hyperparameters used to train the classifier with the PR surrogate objective.

C Detailed Model Selection Analysis

C.1 Overview

To select the generative component for our extraction, categorization, clustering and summarization pipeline, we conducted a systematic evaluation of leading instruction-tuned LLMs that are compatible with local inference via the Ollama runtime. Our goal was to identify a model that provides (i) faithful and context-complete suggestion extraction, (ii) stable semantic similarity judgments for clustering, and (iii) feasibility for deployment on commodity hardware.

This appendix provides the full narrative analysis for each evaluated model, along with a comparison table and a detailed description of the LLM-driven clustering mechanism.

C.2 Model-by-Model Evaluation

Qwen2.5-0.5B-Instruct (with LoRA fine-tuning). We began with Qwen2.5-0.5B-Instruct due to its small footprint and suitability for rapid experimentation. Despite LoRA fine-tuning for suggestion extraction, the model:

- produced vague or incomplete suggestions,
- hallucinated improvement directives not present in the text,
- failed to disambiguate opinionated or descriptive text from actionable suggestions.

Its limited capacity made it unsuitable for downstream clustering or canonicalization.

Qwen1.5 (Quantized, Ollama). This model improved linguistic fluency, but continued to exhibit:

- frequent span selection errors,
- merging of multiple user suggestions into a single incorrect rewrite,
- unstable paraphrasing that reduced cluster cohesion.

Its context window was insufficient for processing dozens of suggestions jointly.

Mistral 7B (Quantized, Ollama). Mistral 7B showed improved stability but suffered from:

- inconsistent extraction fidelity,
- partial or clipped suggestions,
- difficulty recognizing paraphrased suggestions as semantically equivalent,
- limited context capacity for multi-suggestion reasoning.

Llama 2 13B (Quantized, Ollama). This model demonstrated stronger extraction quality than smaller models, but failed to meet clustering requirements:

- similarity judgments were inconsistent across batches,
- clusters were fragmented or over-merged,
- limited context window prevented joint reasoning over large suggestion sets.

Gemma-3-27B (Quantized, Ollama). Gemma-3-27B was the only model that satisfied all requirements:

- reliable, complete, and context-accurate extraction,
- stable paraphrasing without hallucination,
- strong semantic similarity consistency, improving cluster coherence,
- large context window for reasoning over dozens of suggestions simultaneously,
- feasible inference with Ollama Q4_K_M quantization on commodity hardware.

Accordingly, Gemma-3-27B was selected as the final generative model.

C.3 Comparative Model Summary

Table 6 reports the performance of all LLM candidates evaluated during model selection for each pipeline stage.

C.4 Detailed LLM-Based Clustering Mechanism

Clustering in our pipeline is performed entirely using the LLM, without embedding-based or classical clustering algorithms. The process is multi-stage and category-aware:

Step 1: Category-wise Grouping. All extracted suggestions are first grouped according to their assigned category. This ensures that clustering occurs within homogeneous operational domains (e.g., “Food Quality”, “Staff Behavior”).

Step 2: Group Theme Similarity Checks. For each category, the LLM receives pairs of suggestions and determines whether they share the same high-level theme. The decision is based on conceptual similarity rather than lexical overlap (prompt template in Appendix D).

Step 3: Category-Level Clustering. For each category, the LLM processes the *entire set* of extracted suggestions simultaneously. Rather than relying on pairwise similarity scoring, the model performs global theme discovery: it identifies the major conceptual groups that best organize the suggestions in that category.

Model	Extract. Fidelity	Halluc.	Semantic Grouping	Context Window	Hardware Feasible?
Qwen2.5-0.5B	Poor	High	Very Weak	Small	Yes
Qwen1.5	Moderate	High	Weak	Small	Yes
Mistral 7B	Moderate	Moderate	Weak	Small	Yes
Llama 2 13B	Good	Low	Moderate	Small	Yes
Gemma-3 27B	**Excellent**	**Low**	**Strong**	**Large**	**Yes**

Table 6: Comparison of LLM candidates evaluated during model selection.

Step 4: Constructing Clusters. With full visibility of all suggestions, the LLM:

- proposes a coherent set of theme labels (cluster names),
- assigns each suggestion to the most appropriate theme based on conceptual similarity and few-shot clustering rules,
- avoids over-merging by keeping distinct themes separate, and
- **does not force clustering**: suggestions that do not fit any discovered theme are left as standalone items rather than being forced into an incorrect cluster.

This all-at-once, category-level clustering enables holistic reasoning over the entire suggestion set, producing consistent and interpretable clusters while preserving outlier or unique suggestions as individual, actionable items.

Step 5: Cluster Summarization. Each cluster is then summarized by the LLM into short, non-redundant bullet points (see Appendix D for the prompt template).

This LLM-driven clustering method leverages the model's contextual reasoning and large context window, eliminating the need for embeddings or standard clustering algorithms while providing significantly more interpretable outputs.

D Prompt Templates

D.1 Suggestion Extraction Prompt Template

This prompt extracts only explicit, business-directed recommendations from customer reviews.

Components:

- **Role Definition**: Act as an analyst identifying explicit improvement advice.
- **Extraction Criteria**:
  - Must contain a direct advisory or directive expression.
  - Must be explicitly addressed to the business.
  - Must not be inferred or reconstructed.
- **Output Constraints**:
  - Output a single concise paraphrased recommendation.
  - If none exists, output only "NONE".
  - No explanation or commentary.

Abstract Template:

> *"Given a customer review, extract the explicit recommendation addressed to the business, if one is directly stated. Do not infer implied suggestions. If one exists, output a concise paraphrase; otherwise output only 'NONE'."*

D.2 Category Assignment Prompt Template

This prompt assigns each extracted recommendation to a predefined set of operational categories.

Components:

- **Input**: A single recommendation.
- **Category List**: A fixed set of operational categories.
- **Decision Rules**:
  - Assign a category only if a clear correspondence exists.
  - Otherwise return a default "miscellaneous" label.

Abstract Template:

> *"Given a recommendation and a predefined list of category labels, assign the recommendation to the category that best matches its primary theme. If none apply, return a default miscellaneous label. Output only the selected category label."*

D.3 Clustering Prompt Template

This prompt determines whether two recommendations belong to the same broad theme.

Components:

- **Input**: Two recommendations.
- **Task Definition**:
  - Determine whether they address the same operational domain.
  - Focus on broad improvement themes, not lexical similarity.
- **Decision Constraint**: Output one of two labels indicating thematic similarity or dissimilarity.
- **Output**: A single categorical label, no explanation.

Abstract Template:

> *"Given two customer recommendations, determine whether they address the same high-level theme. Consider them similar if they target the same operational area, even if specific actions differ. Otherwise label them as thematically different."*

D.4 Cluster Summarization Prompt Template

Used to generate concise summaries of clustered recommendations.

Components:

- **Input**: A list of related recommendations.
- **Task**: Merge semantically similar items and produce consolidated bullets.
- **Output Requirements**:
  - Bullet-point format.
  - No redundancy.
  - Concise phrasing.
  - Preserve all essential details.

Abstract Template:

> *"Given a set of related customer recommendations, produce concise bullet-point summaries. Merge overlapping items into unified bullets without redundancy. Each bullet should be short, actionable, and capture one coherent improvement suggestion."*

E Dataset Details

We evaluate our system on four held-out test datasets covering two domains. Table 7 represents the data statistics. Test Datasets 1–3 consist of proprietary customer reviews from the restaurant industry and cannot be publicly released. Test Dataset 4 is a publicly available dataset belonging to the ice-cream and frozen-dessert domain. It is sourced from the Yelp Open Dataset (Ice Cream & Frozen Yogurt, Las Vegas, NV), available at: https://business.yelp.com/data/resources/open-dataset/. Review length varies from 1 to 909 tokens (mean 95.5; SD 86.8). All datasets follow the labeling criteria distinguishing business-directed suggestions from general commentary or customer-to-customer advice.

Dataset	Total	0s (Negative)	1s (Positive)
Test Dataset 1 (proprietary)	200	163	37
Test Dataset 2 (proprietary)	200	165	35
Test Dataset 3 (proprietary)	201	164	37
Test Dataset 4	684	591	93

Table 7: Overview of datasets used for testing the classifier.

F RoBERTa's Performance on Test Datasets

Table 8 shows the scores attained by RoBERTa-Base on all the test datasets.

Dataset	Precision	Recall
Test Dataset 1 (proprietary)	0.8919	0.8919
Test Dataset 2 (proprietary)	0.8889	0.9143
Test Dataset 3 (proprietary)	0.9000	0.9730
Test dataset 4	0.9348	0.9094
Average	0.9039	0.9221

Table 8: Precision and Recall scores on test datasets by RoBERTa-Base.

G Cross-Domain Classifier Evaluation

To evaluate generalization beyond the development domains, we tested the RoBERTa-based classifier on four additional industries: real estate, healthcare, finance, and automotive. Each dataset was independently annotated using the same criteria for actionable, business-directed suggestions. Table 9 present the classifier's cross-domain performance and Table 10 provide an overview of the evaluation datasets drawn from additional industry domains.

Industry	Precision	Recall
Real Estate	0.8365	0.9413
Healthcare	0.6887	0.9766
Finance	0.5804	0.9090
Automotive	0.5524	0.9502

Table 9: Cross-domain performance of the classifier on additional industries.

Dataset	Total	0s (Negative)	1s (Positive)
Real Estate	300	269	31
Healthcare	300	287	13
Finance	301	291	09
Automotive	300	283	17

Table 10: Overview of datasets from different industries used for testing the classifier.

Across all domains, recall remained high (0.90–0.98), demonstrating that the classifier generalizes well to unseen industries. Precision varied more substantially, especially in finance and automotive. Manual inspection indicates common sources of false positives include domain-specific terminology (e.g., “APR,” “VIN,” “escrow”), implied or multi-step requests, and procedural narrative styles in healthcare reviews.

H Human Annotation Details

H.1 Annotation Guidelines

All datasets were annotated by trained human annotators following a shared guideline distinguishing explicit business-directed suggestions from general commentary. Annotation was performed at both the review level (for classifier training) and the span level (for extraction evaluation). Disagreements were resolved through majority voting. Annotators were asked to evaluate outputs from four stages of the suggestion pipeline i.e suggestion extraction, category assignment, clustering, and summarization. Each task was rated on a 1–5 Likert scale, where the meaning of scores is shown in Table 11.

Score	Interpretation
1	Very Poor
2	Poor
3	Fair
4	Good
5	Excellent

Table 11: Likert scale used for annotation.

H.2 Suggestion Extraction

Score 5: All suggestions in the review are correctly extracted, with no missing or irrelevant content.

Score 4: Most suggestions are correctly extracted; at most one minor error (missing or extra suggestion).

Score 3: Some suggestions are correctly extracted, but multiple noticeable errors exist.

Score 2: Only a few suggestions are correctly extracted; major errors present.

Score 1: Extraction is unusable or completely incorrect.

H.3 Category Assignment

Score 5: Each suggestion is assigned to the correct category with no errors.

Score 4: Minor categorization mistakes (e.g., 1 misclassified suggestion).

Score 3: Several suggestions assigned incorrectly, but some are correct.

Score 2: Many suggestions misclassified; only a few correct.

Score 1: Nearly all assignments are incorrect or irrelevant.

H.4 Clustering

Score 5: Suggestions within each cluster are highly coherent and semantically similar.

Score 4: Clusters are mostly coherent, with minor inclusion of unrelated suggestions.

Score 3: Some clusters are coherent, but several contain unrelated suggestions.

Score 2: Many clusters contain unrelated or mixed suggestions.

Score 1: Clustering is essentially random or unusable.

H.5 Summarization

Score 5: Summary accurately reflects all main points of the cluster, is fluent, and concise.

Score 4: Summary mostly correct, with minor omissions or phrasing issues.

Score 3: Summary captures some but not all main points; noticeable omissions.

Score 2: Summary inaccurate or misleading, missing most points.

Score 1: Summary unusable or completely irrelevant.

Annotators were instructed to work independently and not discuss ratings during evaluation.

H.6 Raw Annotation Scores

The following tables show the per-annotator scores. The reported values in Table 12 and 13 are the averages across annotators.

Task	Annotator 1	Annotator 2	Annotator 3	Average
Extraction	5	5	5	5.0
Category Assignment	5	4	5	4.6
Clustering	5	5	5	5.0
Summarization	5	4	5	4.6

Table 12: Per-annotator scores for the restaurant dataset.

Task	Annotator 1	Annotator 2	Annotator 3	Average
Extraction	4	4	4	4.0
Category Assignment	4	4	4	4.0
Clustering	5	5	5	5.0
Summarization	5	5	5	5.0

Table 13: Per-annotator scores for the ice-cream shop dataset.

H.7 Annotator Background

Note : The annotators were not involved in model development.

To ensure high-quality evaluation, we worked with three industry experts with extensive experience in handling, labeling, and categorizing customer data across multiple domains. Each annotator has at least over five years of professional experience working with diverse datasets from tens of industries. They are currently employed at a reputed B2B online reputation management company and bring specialized expertise in analyzing customer feedback, sentiment, and suggestions.

All annotators were provided with detailed written guidelines and completed a training phase with practice examples before beginning the actual evaluation. They conducted the annotation independently to minimize bias.

H.8 Inter-Annotator Agreement

Inter-annotator agreement was computed using Fleiss' κ, which adjusts for chance agreement across multiple raters. κ values ranged between 0.74 and 0.85 across tasks, indicating substantial to almost perfect agreement.

I Large Language Model Configuration

All LLM-based components (explicit suggestion extraction, category assignment, clustering, and summarization) use an instruction-tuned and quantized variant of `Gemma-3` deployed through an Ollama runtime. Configuration of the LLM is presented in Table 14.

Property	Value
Model architecture	Gemma-3
Parameter count	27.4B
Quantization	Q4_K_M
Context window	128k tokens
Runtime	Ollama (local inference)

Table 14: LLM configuration used in all generative pipeline components.

J Hardware Configuration

The experiments were conducted on a workstation, Table 15 presents the configurations of the workstation:

Property	Value
GPU Model	NVIDIA RTX A4500
Total VRAM	20,470 MiB

Table 15: Hardware configuration used for training and inference.

K Pipeline Execution Example

Tables 16 and 17 illustrate the execution of the pipeline on real-world review examples, showing the transformation of inputs through each processing stage

Input Review	Label	Extracted Suggestion
Waitress should not have to use their money for the jukebox. Food and service is great!	1	Waitress should not be required to pay for the jukebox.
I like their location. We tried their charcuterie board, lobster soup and steak. My only complaint would be that they have to expand their menu a little to accommodate more vegetarian options.	1	Expand the menu to include more vegetarian options.
Best ice cream in town. All the flavors are great! Mint oreo is my favorite but it's seasonal!	0	NONE (this review was discarded after being labelled 0)
I had the queso empanada for main dish. Our server was also wonderful. I just wish there were a few more vegetarian options for main dishes! Everything else was fantastic!	1	Add a few more vegetarian options for main dishes.
Waited 20 minutes as they were very busy with online orders I think. Please tell customers it will be a wait as some have limited time for lunch. Food was great just service was slow, understand but please notify customer on the wait.	1	Notify customers about potential wait times, especially when busy with online orders.
One of the best chicken I have tasted in a while, nicely seasoned. Loved the crispy fries. Friendly staff. Should add pictures to the menu.	1	Add pictures to the menu.
I called about getting a reservation. The woman told me that if I just walk in though, they could probably seat us pretty quickly. We got there and it was 2 hour wait. Maybe don't tell people you can get them in if you might have a 2 hour wait.	1	Give accurate wait time estimates to customers before they arrive.
Food was really good but had to wait quite a while since they were busy with online orders. Would be nice if they told us about the wait time beforehand.	1	Inform customers about the wait time beforehand.

Table 16: Examples of classification of customer reviews and suggestion extraction.

Extracted Suggestion	Category	Cluster Name	Summarization
Expand the menu to include more vegetarian options.	Menu	Menu Variety & Vegetarian Options	Add more vegetarian options, including main dishes.
Add a few more vegetarian options for main dishes.	Menu	Menu Variety & Vegetarian Options	Add more vegetarian options, including main dishes.
Notify customers about potential wait times, especially when busy with online orders.	Wait Time	Wait Time Communication & Accuracy	Accurately communicate wait times in advance, especially during busy hours.
Give accurate wait time estimates to customers before they arrive.	Wait Time	Wait Time Communication & Accuracy	Accurately communicate wait times in advance, especially during busy hours.
Inform customers about the wait time beforehand.	Wait Time	Wait Time Communication & Accuracy	Accurately communicate wait times in advance, especially during busy hours.
Add pictures to the menu.	Menu	Menu Picture Requests	No summary since this is a standalone suggestion.

Table 17: Examples of category assignment, clustering and summarization of extracted suggestions.