Refining App Reviews: Dataset, Methodology, and Evaluation

Amrita Singh
TCS Research
Pune, India
s.amrita3@tcs.com

Chirag Jain
TCS Research
Pune, India
chirag.rjain3@tcs.com

Mohit Chaudhary
TCS Research
Pune, India
mohit.chaudhary3@tcs.com

Preethu Rose Anish
TCS Research
Pune, India
preethu.rose@tcs.com

Abstract

With the growing number of mobile users, app development has become increasingly lucrative. Reviews on platforms such as Google Play and Apple App Store provide valuable insights to developers, highlighting bugs, suggesting new features, and offering feedback. However, many reviews contain typos, spelling errors, grammar mistakes, and complex sentences, hindering efficient interpretation and slowing down app improvement processes. To tackle this, we introduce RARE (Repository for App review REfinement), a benchmark dataset of 10,000 annotated pairs of original and refined reviews from 10 mobile applications. These reviews were collaboratively refined by humans and large language models (LLMs). We also conducted an evaluation of eight state-of-the-art LLMs for automated review refinement. The top-performing model (Flan-T5) was further used to refine an additional 10,000 reviews, contributing to RARE as a silver corpus.

1 Introduction

The mobile app landscape has seen immense growth, with millions of apps providing essential services (Anthony, 2024). App stores host hundreds of millions of reviews (Ceci, 2022), but only a fraction offer truly informative insights (Noei et al., 2019). User feedback is crucial for developers, offering insights into experiences, bugs, and feature suggestions (Jacek et al., 2022). However, reviews often contain informal language, mixed sentiments, and varied expressions, complicating manual analysis. For example, consider the following review from Spotify *“Love this app! But it crashes all the time. Super frustrating! Fix it plz!”*. This review combines positive feedback with criticism. In addition to this, reviews often include typos, grammatical errors, non-English words, slangs, app-specific jargons and subjective phrases such as *"kinda get boring"* and *"super-addictive"*. Consider another example, from Netflix app review: *"I love netflix but it’s genuinely making me angry that I can’t make my brightness higher bc the app is in control of my brightness panel. So I’m at lunch sitting outside and i can’t see the screen cuz I can’t make the brightness higher because for some reason netflix is in control. It’s frustrating for sure"*. This review clearly expresses user frustration, yet is riddled with informal language(*"cuz"*), irrelevant details (*"So I’m at lunch"*) , non-standard abbreviations (*"bc"*) and unclear statements (*“bc the app is in control of my brightness panel”*), making it difficult for app developers to manually analyze the core concerns of the user. Refining these reviews is essential for enhancing app functionality and improving the overall user experience. While transformer-based language models have excelled in refining natural language text for various downstream tasks such as enhancing code readability (Puri et al., 2021) and question refinement (Liu et al., 2019), app review refinement remains an unexplored area.

We introduce RARE (Repository for App review REfinement), a new benchmark dataset containing 10,000 annotated pairs of original and refined app reviews from 10 different mobile applications. These reviews were generated using state-of-the-art LLMs and the expertise of experienced software engineers. We identified five prevalent issues in app reviews and the necessary operations to rectify them. Using prompt engineering (Reynolds and McDonell, 2021), we designed six prompts to guide GPT-3.5-Turbo (Ye et al., 2023) in refining the reviews. The best-performing prompt was used to generate 10,000 refined reviews, which were then reviewed and corrected by five software engineers with over five years of experience in app development. This formed the gold-standard corpus for RARE.

These original and refined review pairs from

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track, pages 595–608
November 12-16, 2024 ©2024 Association for Computational Linguistics

the gold-standard corpus were used to fine-tune state-of-the-art LLMs for app review refinement, including BART (Lewis et al., 2019), Flan-T5 (Chung et al., 2024), Pegasus (Zhang et al., 2020), Llama-2 (Touvron et al., 2023), Falcon (Almazrouei et al., 2023), Mistral (Jiang et al., 2023), Orca-2 (Mitra et al., 2023), and Gemma (Team et al., 2024). We evaluated these models using human evaluation metrics and standard automatic metrics including, System output Against References and against the Input sentence (SARI) (Xu et al., 2016a), BertScore Precision (BP) (Hanna and Bojar, 2021), Flesch-Kincaid Grade Level FKGL (Kincaid et al., 1975), Flesch-Kincaid Reading Ease (FKRE) (Kincaid et al., 1975), and Average Length (LEN) (Siddharthan, 2014). Flan-T5 emerged as the most effective model based on both human and automatic metrics. We then used Flan-T5 to automatically refine 10,000 additional reviews, creating silver-standard corpus for RARE.

Refined reviews provide several advantages for the app development community: a) Improved user feedback analysis—refined reviews offer developers clearer insights into user sentiments, facilitating better-informed decisions regarding feature enhancements, bug fixes, and user experience improvements. b) Standardization—the refinement process helps standardize the analysis of app reviews. Our preliminary experiments (reported in section 4.2) indicate that refined reviews yield better results in classifying reviews into bug reports, feature requests, and user experience compared to raw reviews. c) Efficient resource allocation—by clarifying user sentiments and common issues, refined reviews enable development teams to allocate resources more effectively, enhancing overall productivity.

The key contributions of our paper are:

1. Identification of five prevalent issues in app reviews and the necessary operations for refinement.

2. Introduction of RARE (Repository for App review REfinement) dataset, featuring 10,000 gold corpus reviews from the Google Play Store and 10,000 silver corpus reviews from the Apple App Store.

3. Experimentation with state-of-the-art transformers to establish baselines for the RARE dataset and a thorough performance evaluation using standard automatic and human metrics.

4. Provision of the RARE dataset and the code for replication purposes in the supplementary material[1]. We believe that RARE can streamline app development by refining user reviews, providing clearer insights, expediting bug fixes and enhancing feature updates.

2 Related Work

Significant efforts have been made on refining natural language text outputs, including summarization (Jusoh et al., 2011), where extracted sentences are refined by omitting unimportant words or phrases before summary generation; content planning (Hua and Wang, 2020), that devises an iterative refinement algorithm to improve incorrectness and incoherence of generated content; questions refinement (Liu et al., 2019), aimed to refine questions by improving readability; and so on ((Hua and Wang, 2020); (Yasunaga and Liang, 2020); (Scheurer et al., 2022); (Du et al., 2022); (He, 2021); (Tsukagoshi et al., 2024); (Ramji et al., 2024)). These works predominantly utilize LLMs to refine text. However, LLMs often face challenges when handling complex text. This difficulty is especially evident in tasks with multifaceted objectives or tasks with hard-to-define goals, such as enhancing program readability (Puri et al., 2021).

In the domains of text simplification and lexical normalization, significant progress has been made, from early rule-based methods (Chandrasekar and Srinivas, 1997) to statistical models ((Zhu et al., 2010); (Coster and Kauchak, 2011); (Kauchak, 2013); (Hwang et al., 2015); (Xu et al., 2016b)). The introduction of transformer-based models such as BERT and GPT has advanced the field, achieving top results in domains such as medical, legal, clinical, news, and Wikipedia texts ((Jiang et al., 2020); (Li et al., 2022); (Van et al., 2020); (Joseph et al., 2023)). We acknowledge that Simplification and refinement are related concepts and can overlap in some cases, but their goals are different. While simplification aims to make text easier to understand, refinement in our context focuses on making reviews more actionable for developers by ensuring clarity, removing irrelevant details, and maintaining technical accuracy.

The works reported above primarily focus on refining outputs based on a single objective. In contrast, our task of refining app reviews

[1] https://zenodo.org/records/13939427

encompasses multiple facets, including ensuring grammatical accuracy, rephrasing, removing irrelevant words and information, rearranging words and information, and modifying sentences. These tasks necessitate careful handling due to their nuanced and diverse requirements. Furthermore, none of the existing literature specifically addresses app review refinement. To the best of our knowledge, our work represents the first effort in the area of app review refinement.

3 RARE: A New Benchmark Dataset

Due to the absence of a ground truth for automated app review refinement, we created the RARE (Repository for App review REfinement) dataset. RARE includes 10,000 annotated pairs of raw and refined reviews as the gold corpus, and an additional 10,000 reviews refined by the best-performing model as the silver corpus. This dataset benchmarks LLMs and other machine learning models, aiding future research in automated app review refinement. Figure 1 provides an overview of our dataset collection, analysis, and refinement process.

3.1 Data Extraction

We collected 1,000 reviews per app from 10 different apps, resulting in 10,000 reviews from the Google Play Store and 10,000 from the Apple App Store (20,000 reviews in total). These domains included *Communication (WhatsApp), Travel (Uber), Music & Audio (Spotify), social media (Twitter), Video Player & Editor (YouTube), Entertainment (Netflix), Games (Candy Crush Saga), Shopping (Amazon), Education (Duolingo),* and *Health (Google Fit).* The reviews were extracted based on the following criteria: (1) over 10 words; (2) written in English; and (3) starting from the most recent. Notably, despite the platform differences, we observed no significant variation in review patterns between the Google Play and Apple App stores. After extraction, 10,000 raw reviews from the Google Play Store and 10,000 raw reviews from the Apple App Store were saved in separate Excel files including the app name and review. These extracted reviews served as the raw reviews for the gold and silver corpora in the RARE dataset. The algorithm summarizing the extraction process is presented in Appendix A.

3.2 Collaborative Review Refinement Process

In this section, we outline the collaborative review refinement process involving software engineers and LLMs.

The first three authors manually analyzed 500 raw reviews to identify the prevalent issues in app reviews and the corresponding corrective operations. Five operations were identified. Based on these operations, six prompts were designed for refining the raw reviews (see section 3.2.1). During the pilot refinement phase, GPT-3.5-Turbo generated 3000 refined reviews using these prompts (500 reviews per prompt) (see section 3.2.2). A quantitative analysis identified the best prompt based on automated and human assessments (see section 3.2.2.1). Further qualitative analysis was performed, with five software engineers manually refining reviews as needed (see section 3.2.2.2). Insights from this phase helped establish the RARE benchmark dataset, comprising 10,000 gold corpus and 10,000 silver corpus refined reviews (see section 3.2.3). Each step is detailed in the subsections below.

3.2.1 Operations Identification and Prompt Generation

We conducted a manual analysis on a randomly selected set of 50 reviews per app, totaling 500 raw reviews. The first three authors independently read each raw review to identify issues that might hinder comprehension. The authors then worked together to reach a consensus regarding the identified issues and the operations to address them. Five key operations were determined: *Grammatical Accuracy, Rephrasing, Deleting Irrelevant Words and Information, Rearranging Words and Information, Sentences Operations.* Detailed information about these operations is provided below:

Grammatical Accuracy: Correcting grammar errors, such as typos, spelling mistakes, and punctuation issues present in the raw reviews.

Rephrasing: Modifying complex, ambiguous, and difficult-to-understand words and phrases from the raw reviews with simpler alternatives in the refined reviews while maintaining their original meaning.

Deleting Irrelevant Words and Information: Removing extraneous text from the raw review to make it clear and concise while preserving the original meaning and tone.

Rearranging Words and Information: Organizing the words and information within a raw review

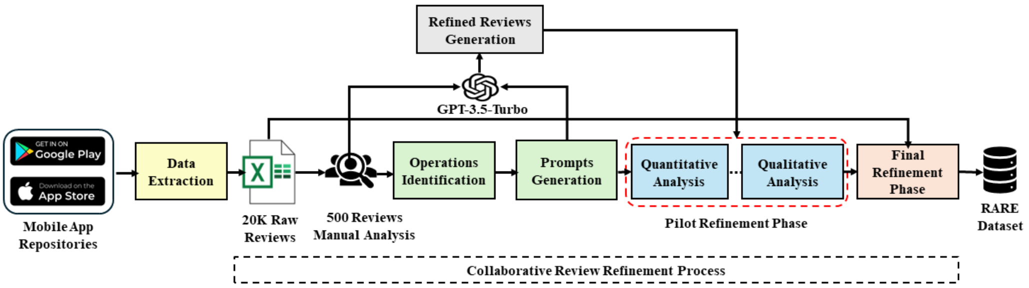


Figure 1: Overview of RARE Dataset Creation

sentence into a logical and easily followed order to enhance comprehension and flow.

Sentences Operations: Applying techniques such as breaking down complex sentences (sentence splitting) and reordering raw review sentences to enhance clarity and readability.

The decision to apply these operations was influenced by factors such as complexity, ambiguity, and overall readability of the raw reviews. Our goal was to simplify wording for improved clarity, particularly to avoid confusion for software engineers caused by abbreviations or complex phrasing. While some changes may seem subtle, we aimed for consistent clarity across reviews.

Our analysis revealed that rephrasing was the most commonly needed operation, required in 35% of cases, followed by grammatical corrections (24%), deletion of irrelevant information (28%), rearrangement of content (25%), and sentence restructuring (27%). The aforementioned operations contribute to enhancing overall text comprehension (Action and Network, 2011) and reducing the cognitive effort required to understand the text (Chamovitz and Abend, 2022).

To execute these operations on raw app reviews, we employed prompt engineering techniques (Reynolds and McDonell, 2021) and designed six distinct prompts to guide GPT-3.5-Turbo in refining raw reviews while preserving the original meaning and user intent. While designing the prompts, we included both the content (instructions given to refine the reviews, such as ensuring clarity, conciseness, and brevity) and the context in which they are specifically applied (app review refinement in our case). Our experiments showed that clear and concise prompts produced better results, while overly detailed ones caused confusion. The guidelines to design the six prompts and a comprehensive list of prompts are provided in Appendix A.

3.2.2 Pilot Refinement Phase

In the pilot refinement phase, each of the six prompts was used with 500 raw reviews to guide GPT-3.5-Turbo in generating refined versions of the raw reviews, resulting in 3000 refined reviews (500 per prompt). We then conducted quantitative and qualitative analysis to assess their quality.

3.2.2.1 Quantitative Analysis

For the quantitative analysis, we computed several standard metrics, including automatic metrics: FKGL, FKRE, LEN, and Similarity Score (SS) (Rahutomo et al., 2012), alongside human metrics: Q_a, Q_b, Q_c and Q_d (Sulem et al., 2018). The response options for the human metrics were: 1 ("*No*"), 2 ("*Maybe*"), and 3 ("*Yes*"). Detailed information about each metric is provided in Table 1. Evaluation using human metrics was conducted by five software engineers with over 5 years of experience in app development. As a part of their job profile, these software engineers often dealt with user reviews received on their apps. Their job involved reading and comprehending the raw reviews, manually finding the bugs reports, feature requests and usability issues, prioritizing them and then making app enhancement decisions. We distributed 3000 refined reviews equally among these software engineers, ensuring that each of the five software engineers received 600 reviews (100 refined reviews generated from each prompt). The results of the evaluation using these metrics are presented in Table 2.

From Table 2, it is evident that the output from *Prompt1* demonstrates good results in grammatical refinement (Q_a) and in preserving intended meaning (Q_b & Q_c). However, it lacks simplification (Q_d), resulting in a higher grade level required to comprehend the text (FKGL) compared to the raw review. Additionally, readability is

Evaluation	Metric	Definition
Human Metrics	Q$_a$	Is the refined review fluent and grammatically correct?
	Q$_b$	Does the refined review add any irrelevant information that was not present in the raw review?
	Q$_c$	Does the refined review remove any important information that was present in the raw review?
	Q$_d$	Is the refined review easier to understand when compared with the raw review?
Automatic Metrics	FKGL	It measures text complexity using sentence length and syllable count, with lower scores indicating simpler text.
	FKRE	It evaluates text readability based on average sentence length and average number of syllables per word. A higher score indicates easier readability.
	LEN	It measures the average length of the sentence.
	SS	It assesses how closely the meanings of two texts align using cosine similarity, where a score nearing 1 indicates strong similarity.
	SARI	It evaluates how well the output sentence aligns with the reference and input sentence. Higher SARI score indicates better sentence simplification quality, while a lower score indicates poorer performance.
	BP	It evaluates machine-generated text by comparing it to a reference, focusing on how well it maintains the original meaning. Higher precision signifies better alignment in word choice and semantics.

Table 1: Metric Overview

reduced compared to the raw review, as indicated by lower reading ease scores (FKRE) and relatively low similarity scores (SS). *Prompt2* shows improvement in simplification (Q$_d$) and achieves a good average length (LEN). However, it performs poorly in preserving intended meaning (Q$_b$ & Q$_c$) and readability (FKRE). *Prompt3* maintains high scores in grammatical refinement (Q$_a$) and meaning preservation (Q$_b$ & Q$_c$), but it lacks in making the text more accessible and simpler (Q$_d$), affecting readability negatively (FKRE). *Prompt4* improves in simplification (Q$_d$) and grammatical refinement (Q$_a$), but it exhibits higher complexity (FKGL) compared to the raw review, indicating issues with sentence structure and vocabulary, and struggles in retaining important information (Q$_c$). Both *Prompt5* and *Prompt6* demonstrate optimal values across most standard metrics. However, a comparison reveals that *Prompt6* outperforms *Prompt5* in grammatical refinement (Q$_a$), simplification (Q$_d$), and overall similarity score (SS). Additionally, *Prompt6* maintains a good balance in preserving meaning (Q$_b$ & Q$_c$), grade level (FKGL), readability score (FKRE), and achieves an optimal length (LEN). Therefore, *Prompt6* is selected as the optimal prompt for generating refined reviews in the final refinement phase. Using a single prompt (*Prompt6*) for refining all the 10,000 reviews ensured consistency, saved time and resources, and reduced variability. This approach allowed for clearer benchmarking and more practical management of large datasets.

3.2.2.2 Qualitative Analysis

After conducting quantitative analysis, we determined that the refined reviews (*Refined6*) generated by *Prompt6* were superior compared to those generated by other prompts. Subsequently, the refined reviews (*Refined6*) underwent further validation. They were evenly distributed among five software engineers (who performed evaluations using human metrics during the quantitative analysis) for manual inspection and corrections. These software engineers identified specific errors that needed correction in the refined reviews (*Refined6*) produced by GPT-3.5-Turbo, as outlined below:

Deletion of Relevant Words and Information: This issue occurs when GPT-3.5-Turbo fails to accurately differentiate between essential and non-essential information during review refinement, leading to the omission of critical details. For example, the raw review, '*it's showing EMI value in bold numbers instead of showing the actual price,*' got refined to '*price is shown as bold numbers,*' omitting the crucial word '*EMI*'. Such omissions can misguide readers, as they may not recognize that users are referring to EMI values instead of actual prices, thereby impacting the quality and completeness of the refined review.

Addition of Superfluous Words and Information: This issue occurs when GPT-3.5-Turbo adds unnecessary words or information during review refinement, altering the review's intended context and clarity. For example, the raw review, "*Doesn't even update the data even after I put an activity,*" got refined to "*It also doesn't update the data even after I add an activity. This causes frustration and inconvenience,*" introducing sentiments of frustration and inconvenience not explicitly mentioned in the original review.

Oversimplification Leading to Ambiguity: This issue arises when GPT-3.5-Turbo overly simplifies information, failing to convey the intended meaning or depth. Consequently, the output may not fully capture the intricacies of the context,

Input		Output	Automatic Metric				Human Metric			
Prompt	Review		FKGL↓	FKRE↑	LEN↓	SS↑	Qa↑	Qb↑	Qc↑	Qd↑
None	Raw	Raw	6.46	**76.47**	15.57	—	—	—	—	—
Prompt1		Refined1	8.96	54.83	13.48	94.16	**99.2**	**99.73**	**99.46**	90.66
Prompt2		Refined2	**6.37**	71.52	**12.44**	94.04	97.46	96.13	87.6	**93.46**
Prompt3		Refined3	7.11	69.63	14.33	94.9	**98.39**	**99.46**	**98.13**	91.73
Prompt4		Refined4	7.01	70.11	14.19	**95.19**	**99.46**	99.2	95.46	**93.33**
Prompt5		Refined5	**5.37**	**80.22**	**13.27**	**95.49**	96.53	97.86	**98.39**	90.4
Prompt6		Refined6	**5.36**	**80.06**	**13.17**	**95.67**	97.86	**99.33**	97.73	**92.53**

Table 2: Results of quantitative analysis where red highlights denote the first-best value, blue highlights denote the second-best value, and green highlights denote the third-best value.Please note that an upward arrow (↑) in the headings signify 'higher is better', while a downward arrow (↓) signify 'lower is better'.

leading to ambiguity. For example, a raw review mentions, "*you are turning your free features into premium - 1. Play in order 2. Normal shuffle 3. Lyrics in few songs 4. Queue list 5. List view 6. Seek movement 7. Replay/ Loop 8. Previous song 9. Limited skips to next song.*" GPT-3.5-Turbo refines this to "*some features that used to be free are now only available with a premium subscription.*" The oversimplification makes it unclear which specific features the user is referring to. The qualitative analysis revealed that deletion of relevant words occurred in around 9% of refined reviews, the addition of unnecessary words in 5%, and oversimplification leading to ambiguity in 11% of refined reviews.

These issues highlight GPT-3.5-Turbo's lack of necessary domain-specific knowledge for accurately refining raw reviews. Additionally, employing GPT-3.5-Turbo to process a large volume of reviews is impractical due to high costs. Given these limitations, the five software engineers manually corrected the refined reviews (*Refined6*). This manual refinement process typically required 0.5 to 1 minute per review, significantly less than the 4 to 5 minutes needed to manually write a refined review from scratch.

3.2.3 Final Refinement Phase

In the final refinement phase, we selected 9,500 raw reviews from the Google Play Store, comprising 950 reviews from each of 10 different apps. These reviews were refined through collaboration involving GPT-3.5-Turbo and software engineers, detailed in Section 3.2.2. This process created a gold corpus within RARE dataset, with 10,000 annotated review pairs (9,500 refined in the final phase and 500 in the pilot phase). Subsequently, this corpus was used to fine-tune eight state-of-the-art models (Section 4.1). The best performing model, Flan-T5 (Section 4.2), refined an additional 10,000 raw reviews from the Apple App Store, forming a silver corpus within RARE. Statistics of the RARE dataset are provided in Appendix B. Additionally, Table 6 in Appendix B presents a few examples of raw and refined reviews from both the gold and silver corpora.

4 Experiments

4.1 Baseline Models

We evaluated eight transformer-based models known for their state-of-the-art performance in NLP: BART, Flan-T5, Pegasus, Llama-2, Falcon, Mistral, Orca-2, and Gemma. The gold corpus was split into two sets: 5000 reviews (500 from each of the 10 apps) for training and another 5000 reviews for testing. These models were fine-tuned on the training set and used to generate refined reviews for the testing set. Additionally, to set a baseline, we also experimented using normalization technique, specifically stemming, on the raw reviews. Performance results are detailed in Table 3 and Table 4 (Section 4.2), with hyperparameters provided in Appendix A.

4.2 Results and Discussion

Automatic Evaluation

Table 3 presents the performance of the baseline models. Flan-T5 stands out as the top model for app review refinement, achieving the highest BP score of 94.26, indicating superior preservation of review meaning compared to others. It also performs well across SARI, FKGL, FKRE, and LEN metrics, making it the optimal choice. BART follows closely with a high BP score and FKRE, but produces longer reviews on an average. Orca-2 excels in SARI but lags in BP, suggesting less robust meaning preservation. Falcon generates the shortest reviews, but compromises on BP. Gemma ranks highest in FKGL but lowest in BP, indicating compromised meaning. Pegasus scores well in BP but sacrifices simplification. Mistral and Llama-2 show good BP and SARI scores but lower FKGL and FKRE scores, impacting readability. The baseline model (normalization), which employs stemming, demonstrated the lowest SARI score and the longest review length. These findings suggest that normalization employing stemming is less effective in refining app reviews, highlighting its inadequacy in addressing complex refinement tasks. In summary, Flan-T5 stands out as the optimal model, excelling in meaning preservation, simplification, readability, and conciseness. Although the other models exhibit various strengths, each has limitations in one or more areas.

Reviews	**Model**	**Metrics**				
		SARI↑	**BP↑**	**FKGL↓**	**FKRE↑**	**LEN↓**
Raw	__	__	__	7.2	73.83	17.08
Refined	BART	54.33	**93.88**	5.1	**81.77**	13.07
	Flan-T5	**55.2**	**94.26**	**5.01**	**81.37**	12.48
	Pegasus	51.26	**93.63**	5.07	**81.49**	12.81
	Llama-2	54.08	92.1	5.28	78.49	11.97
	Falcon	54.03	88.17	**4.83**	79.85	**10.9**
	Mistral	53.61	92.79	5.32	78.65	12.21
	Orca-2	**57.6**	88.78	5.12	79.33	**11.79**
	Gemma	**55.98**	81.41	**4.78**	80.5	**11.08**
	Normalization Baseline	37.51	86.7	6.16	80.97	17.08

Table 3: Results of the eight baseline models using automatic metrics where red highlights denote the first-best value, blue highlights denote the second-best value, and green highlights denote the third-best value

Model	**Q_a↑**	**Q_b↑**	**Q_c↑**	**Q_d↑**
BART	**95**	**97.33**	94.33	**89.67**
Flan-T5	**95.67**	**98.33**	**95.67**	**90.33**
Pegasus	89	78.67	92	74
Llama-2	**94.67**	**95.33**	92	**81.67**
Falcon	94	89.67	**95.33**	80
Mistral	93.67	82.33	93.33	80
Orca-2	88.67	86	94	79.33
Gemma	90.33	83	**94.67**	78.33
Normalization Baseline	67.42	91.27	90.35	71.23

Table 4: Results of the eight baseline models using human metrics where red highlights denote the first-best value, blue highlights denote the second-best value, and green highlights denote the third-best value

Human Evaluation

Due to the resource-intensive nature, manual evaluation of the entire testing set for each model was impractical. Hence, a subset of 100 reviews refined by each model underwent human evaluation using metrics Q_a, Q_b, Q_c and Q_d by the five software engineers from the pilot refinement phase. The results presented in table 4 clearly indicate that Flan-T5 stands out as the best-performing model for app review refinement, even in terms of human metrics. The BART model closely follows Flan-T5, as evidenced by both automatic and human evaluations.

Although our dataset consists of reviews from only 10 mobile applications, we ensured representation across diverse domains, including *Communication, Travel, Music & Audio, Social Media, Video Player & Editor, Entertainment, Games, Shopping, Education,* and *Health*. This diversity enabled us to capture a wide array of user experiences and review types, contributing to the generalizability of our model. While we acknowledge that a larger dataset could enhance the model's robustness and accuracy, the breadth of domains included in our current dataset provides a comprehensive view of varied user sentiments and contexts.

To demonstrate the benefits of app review refinement for the broader app development community, we conducted a small-scale multi-label classification task on 1,000 reviews, categorizing them into bug reports, feature requests, and user experience insights. The results revealed a weighted average F1 score of 0.81 for raw reviews and an improved score of 0.89 for refined reviews, indicating the significant value added by the refinement step.

5 Conclusions and Future Work

In this work, we introduce RARE, a benchmark for App Review Refinement. RARE comprises a corpus of 10,000 annotated reviews, collaboratively refined by humans and LLMs sourced from 10 different application domains, constituting the gold corpus. Additionally, it includes a set of 10,000 automatically refined reviews, forming the silver corpus. We evaluated eight state-of-the-art models and determined that Flan-T5 is the best-performing model for app review refinement. The complete RARE benchmark and code are included in the supplementary material, establishing RARE as a benchmark in text refinement for app development. Future work may focus on two directions: First, extracting non-functional requirements from app reviews and assessing how app review refinement enhances this process compared to using raw reviews; second, summarizing the extracted requirements from these app reviews.

References

Plain Language Action and Information Network. 2011. *Federal plain language guidelines*. CreateSpace Independent.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, et al. 2023. The falcon series of open language models. *arXiv preprint arXiv:2311.16867*.

James Anthony. 2024. Number of apps in leading app stores in 2024: Demographics, facts, and predictions.

Laura Ceci. 2022. Global reviews of top android apps by category 2022.

Eytan Chamovitz and Omri Abend. 2022. Cognitive simplification operations improve text simplification. *arXiv preprint arXiv:2211.08825*.

Raman Chandrasekar and Bangalore Srinivas. 1997. Automatic induction of rules for text simplification. *Knowledge-Based Systems*, 10(3):183–190.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

William Coster and David Kauchak. 2011. Simple english wikipedia: a new text simplification task. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 665–669.

Wanyu Du, Zae Myung Kim, Vipul Raheja, Dhruv Kumar, and Dongyeop Kang. 2022. Read, revise, repeat: A system demonstration for human-in-the-loop iterative text revision. *arXiv preprint arXiv:2204.03685*.

Michael Hanna and Ondřej Bojar. 2021. A fine-grained analysis of bertscore. In *Proceedings of the Sixth Conference on Machine Translation*, pages 507–517.

Xingwei He. 2021. Parallel refinements for lexically constrained text generation with bart. *arXiv preprint arXiv:2109.12487*.

Xinyu Hua and Lu Wang. 2020. Pair: Planning and iterative refinement in pre-trained transformers for long text generation. *arXiv preprint arXiv:2010.02301*.

William Hwang, Hannaneh Hajishirzi, Mari Ostendorf, and Wei Wu. 2015. Aligning sentences from standard wikipedia to simple wikipedia. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 211–217.

Dąbrowski Jacek, Letier Emmanuel, Anna Perini, and Angelo Susi. 2022. Analysing app reviews for software engineering: a systematic literature review. *Empirical Software Engineering*, 27(2).

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Chao Jiang, Mounica Maddela, Wuwei Lan, Yang Zhong, and Wei Xu. 2020. Neural crf model for sentence alignment in text simplification. *arXiv preprint arXiv:2005.02324*.

Sebastian Joseph, Kathryn Kazanas, Keziah Reina, Vishnesh J Ramanathan, Wei Xu, Byron C Wallace, and Junyi Jessy Li. 2023. Multilingual simplification of medical texts. *arXiv preprint arXiv:2305.12532*.

Shaidah Jusoh, Abdulsalam M Masoud, and Hejab M Alfawareh. 2011. Automated text summarization: sentence refinement approach. In *International Conference on Digital Information Processing and Communications*, pages 207–218. Springer.

David Kauchak. 2013. Improving text simplification language modeling using unsimplified text data. In *Proceedings of the 51st annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 1537–1546.

J Peter Kincaid, Robert P Fishburne Jr, Richard L Rogers, and Brad S Chissom. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Jiazhao Li, Corey Lester, Xinyan Zhao, Yuting Ding, Yun Jiang, and VG Vydiswaran. 2022. Pharmmt: a neural machine translation approach to simplify prescription directions. *arXiv preprint arXiv:2204.03830*.

Ye Liu, Chenwei Zhang, Xiaohui Yan, Yi Chang, and Philip S Yu. 2019. Generative question refinement with deep reinforcement learning in retrieval-based qa system. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1643–1652.

Arindam Mitra, Luciano Del Corro, Shweti Mahajan, Andres Codas, Clarisse Simoes, Sahaj Agarwal, Xuxi Chen, Anastasia Razdaibiedina, Erik Jones, Kriti Aggarwal, et al. 2023. Orca 2: Teaching small language models how to reason. *arXiv preprint arXiv:2311.11045*.

Ehsan Noei, Feng Zhang, and Ying Zou. 2019. Too many user-reviews! what should app developers look at first? *IEEE Transactions on Software Engineering*, 47(2):367–378.

Ruchir Puri, David S Kung, Geert Janssen, Wei Zhang, Giacomo Domeniconi, Vladimir Zolotov, Julian Dolby, Jie Chen, Mihir Choudhury, Lindsey Decker, et al. 2021. Codenet: A large-scale ai for code dataset for learning a diversity of coding tasks. *arXiv preprint arXiv:2105.12655*.

Faisal Rahutomo, Teruaki Kitasuka, Masayoshi Aritsugi, et al. 2012. Semantic cosine similarity. In *The 7th international student conference on advanced science and technology ICAST*, volume 4, page 1. University of Seoul South Korea.

Keshav Ramji, Young-Suk Lee, Ramón Fernandez Astudillo, Md Arafat Sultan, Tahira Naseem, Asim Munawar, Radu Florian, and Salim Roukos. 2024. Self-refinement of language models from external proxy metrics feedback. *arXiv preprint arXiv:2403.00827*.

Laria Reynolds and Kyle McDonell. 2021. Prompt programming for large language models: Beyond the few-shot paradigm. In *Extended abstracts of the 2021 CHI conference on human factors in computing systems*, pages 1–7.

Jérémy Scheurer, Jon Ander Campos, Jun Shern Chan, Angelica Chen, Kyunghyun Cho, and Ethan Perez. 2022. Training language models with language feedback. *arXiv preprint arXiv:2204.14146*.

Advaith Siddharthan. 2014. A survey of research on text simplification. *ITL-International Journal of Applied Linguistics*, 165(2):259–298.

Elior Sulem, Omri Abend, and Ari Rappoport. 2018. Semantic structural evaluation for text simplification. *arXiv preprint arXiv:1810.05022*.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Hayato Tsukagoshi, Tsutomu Hirao, Makoto Morishita, Katsuki Chousa, Ryohei Sasano, and Koichi Takeda. 2024. Wikisplit++: Easy data refinement for split and rephrase. *arXiv preprint arXiv:2404.09002*.

Hoang Van, David Kauchak, and Gondy Leroy. 2020. Automets: the autocomplete for medical text simplification. *arXiv preprint arXiv:2010.10573*.

Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. 2016a. Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics*, 4:401–415.

Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. 2016b. Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics*, 4:401–415.

Michihiro Yasunaga and Percy Liang. 2020. Graph-based, self-supervised program repair from diagnostic feedback. In *International Conference on Machine Learning*, pages 10799–10808. PMLR.

Junjie Ye, Xuanting Chen, Nuo Xu, Can Zu, Zekai Shao, Shichun Liu, Yuhan Cui, Zeyang Zhou, Chao Gong, Yang Shen, et al. 2023. A comprehensive capability analysis of gpt-3 and gpt-3.5 series models. *arXiv preprint arXiv:2303.10420*.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International conference on machine learning*, pages 11328–11339. PMLR.

Zhemin Zhu, Delphine Bernhard, and Iryna Gurevych. 2010. A monolingual tree-based translation model for sentence simplification. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 1353–1361.

A Prompts and Hyperparameters

We first summarize the app review data extraction process in Algorithm 1 and then provide the six prompts (detailed in Figure 2) that were used during the pilot refinement phase. These prompts were employed to generate refined reviews by prompting GPT-3.5-Turbo. The guidelines to design six prompts are detailed below:
Prompt 1: Brief and clear with simple instructions.
Prompt 2: Prioritized clarity but was lengthy.
Prompt 3: Provided a comprehensive task description.
Prompt 4: Presented as a mathematical expression.
Prompt 5: Guided the model iteratively with brief instructions.
Prompt 6: Similar to Prompt 5 but with additional details.

Next, the specifications of hyperparameters and configurations utilized by transformer-based models in the experiments are given in Table 5. Grid search technique was used to optimize these hyper-parameter values.

Algorithm 1 Review Extraction Algorithm

Input: Application domain set D, store set S, reviews per store per domain R_{sd}, recent review size R_{size}
$L(r)$ be a function that returns length of reviews
$E(r)$ be a function that returns 1 if review r is in English, 0 otherwise
Output: Extracted review set

```
Initialize reviews_data = { }
for each store $s$ in $S$ do
    for each application domain $d$ in $D$ do
        reviews = mostRecentReviews ($R_{size}$)
        review = { $r \in$ reviews & $L(r) > 10$ & $E(r) = 1$ }
        reviews_data = reviews_data ∪ review
        if len(reviews_data) $\geq R_{sd}$ then
            break
```

Model	Hyperparameters
BART	per_device_train_batch_size=1
	num_train_epochs = 1
	learning_rate=3e-5
	weight_decay=0.01
	save_steps=500
	save_total_limit=3
Flan-T5 and Pegasus	per_device_train_batch_size=1
	num_train_epochs = 1
	learning_rate=5.6e-5
	weight_decay=0.01
	save_steps=500
	save_total_limit=3
Llama-2	num_train_epochs=8
	per_device_train_batch_size=4
	gradient_accumulation_steps=1
	optim="paged_adamw_32bit"
	save_steps=500
	learning_rate=2e-4
	weight_decay=0.001
Falcon	num_train_epochs=8
	per_device_train_batch_size=4
	gradient_accumulation_steps=4
	optim="paged_adamw_32bit"
	save_steps=500
	learning_rate=2e-4
	weight_decay=0.001
Mistral	num_train_epochs=8
	per_device_train_batch_size=2
	gradient_accumulation_steps=1
	optim="paged_adamw_32bit"
	save_steps=500
	learning_rate=3e-4
	weight_decay=0.001
Gemma and Orca-2	num_train_epochs=4
	per_device_train_batch_size=2
	gradient_accumulation_steps=1
	optim="paged_adamw_32bit"
	save_steps=500
	learning_rate=3e-4
	weight_decay=0.001

Table 5: Hyper-parameters details of each model

B Statistics of the RARE Dataset

In Appendix B, we first present examples of raw and refined reviews from the gold and silver corpus in Table 6. Next, we display the word count distribution of the gold and silver corpus in Figure 3. From Figure 3, it can be observed that the refined reviews are typically 20-25% shorter than original reviews, ensuring clarity and conciseness while preserving the key information.

Following this, we present the word cloud distribution of the gold and silver corpus in Figure 4. From Figure 4, it can be observed that there is a noticeable reduction in irrelevant or noisy terms present in the raw reviews, suggesting that the refinement process enhances the quality and relevance of the words within both corpus.

Finally, we show the FKGL distribution of each app from the gold and silver corpus in Figure 5. From Figure 5, it is clear that the readability of refined reviews for both corpus is significantly better compared to the raw reviews.

Prompt 1

Please create a refined version of the app review, ensuring it adheres to the following criteria:

1. The refined review must be free of typos, spelling errors, punctuation issues, and grammatical mistakes.

2. The refined review should be concise, while retaining all details from the app review, maintaining its original meaning.

3.The refined review should be short, clear, and easy to read and understand.

4. The refined review should avoid adding any new information that is not present in the app review and should also avoid unnecessary rephrasing.

Here is the app review: {X}

Prompt 2

You will be provided with an app review. Your task is to refine the review iteratively until it achieves a clarity score between 4 to 5. Each iteration involves refining the review and evaluating its clarity. Please make sure to follow the evaluation steps and criteria carefully.
App Review: {X}
Evaluation Steps:
1. Carefully read the app review to grasp its key points and details.
2. Refine the review while retaining its main message and essential details.
3. Ensure the refined review is free of typos, spelling errors, punctuation issues, and grammatical mistakes.
4. Keep the refined review short, clear, easy to read, and understand.
5. Evaluate the clarity of the refined version based on the provided scale.
6. If the clarity score falls within the range of 4 to 5, stop the iteration. Otherwise, continue refining and evaluating until the desired clarity score is achieved.
Evaluation Criteria:
Clarity Score (1-5) - how effectively the refined version retains the main message and important details of the review while making it easier to understand. Effective refinement should convey the essence of the review without losing crucial information, avoiding becoming vague or losing its original meaning due to excessive refinement.

Prompt 3

Please create a refined version of the app review, ensuring it adheres to the following criteria: Refine the given app review enclosed in triple back ticks using the below steps:
App Review: ``` {X} ```
Steps:
1. Analyze the review text to understand its content and context.

2. Identify technical and non-technical terms that are difficult to understand and simplify them.

3. Break down complex sentences and replace technical terms with simpler alternatives to enhance clarity while maintaining conciseness.

4. Make it flawless and grammatically correct while preserving the app review voice and sentiment, avoiding significant meaning shifts or changes in tone.

5. Review the refined text to ensure it flows naturally, conveys the same message, and is easier to understand.

6. Repeat the above steps if needed.

7. Output the refined review once it meets the specified criteria.

Prompt 4

R = ```{X}```

F(x) = Refine

g = Make it grammatically correct and flawless.

c = Concise and clear.

j = Simplify jargons and avoid verbosity.

v = Preserve original voice and sentiment, retain information from app review.

Factors = g && c &&j && v

Y = F(R, factors)

Y =>

Prompt 5

Given an app review, your task is to create a refined version that is easily understandable by a general audience. Following the below steps:

App Review: {X}

Steps:

1. Read the review and try to replace complex words with simpler alternatives.

2. Shorten sentences and use straightforward tones like active voice.

3. Ensure each sentence has a clear main idea and has no flaws.

4. Clarify pronoun references for coherence.

5. Maintain consistent terminology throughout the text and aim for a readability level suitable for the intended audience.

6. Generate refined output and check whether generated output has good readability, if not, then repeat above steps.

Prompt 6

Given an app review with intricate sentence structures, dependent clauses, and nuanced vocabulary, your task is to generate a refined version of the text.

App Review: {X}

Focus on the following steps to ensure refinement:
Steps:
1. Restructure complex sentences into simpler ones with clear subject-verb relationships.

2. Simplify or break down dependent clauses to enhance clarity and readability.

3. Substitute complex or technical terms with simpler alternatives, avoiding jargon whenever possible, and having no flaws.

4. Aim for brevity while retaining the core meaning of the text.

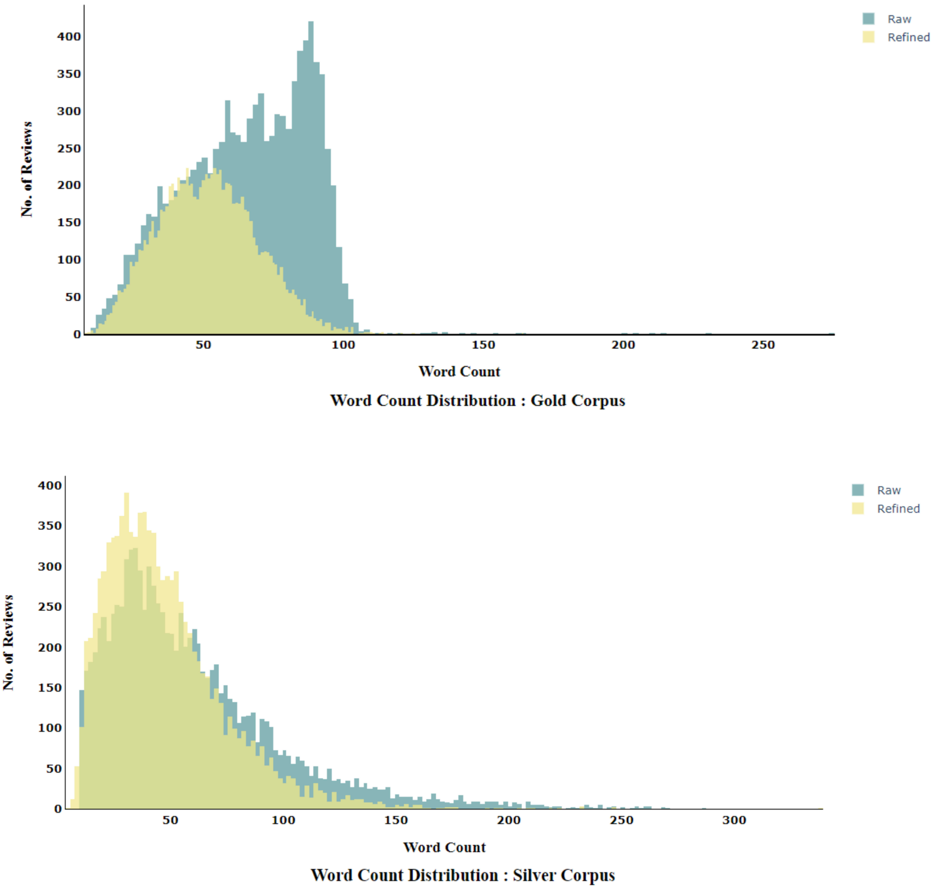
5. Ensure that the refined text maintains logical flow and coherence, preserving the main ideas from the original and make it properly readable.

6. Generate refined output and check whether generated output has good readability, if not, then repeat above steps.

Figure 2: Prompts used by GPT-3.5-Turbo for refining raw reviews

Gold Corpus	**Raw**	Uber hire anybody literally i request a ride and the driver is nearby i kept calling him he doesn't answer kept texting him and he doesn't answer either and basically he stayed in the same place and when i turned off live location he came close to the pickup point but not to the actual pickup point and then cancelled the trip that way he can put fee on me as like i didn't show but it was actually him uber used to be providing with good service but not anymore I'll be using other apps
	Refined	Uber hired a driver who didn't answer my calls or texts when he was nearby for my ride. He stayed in one place and only moved closer to the pickup point when I turned off live location. Then, he canceled the trip, charging me a fee as if I didn't show up. Uber used to provide good service, but not anymore. I'll use other apps instead.
	Raw	This app I very good but the only huge problem I'm encountering is when I'm trying to record a video and edit it I want to put the video as high quality as possible but it doesn't registered like the old ones for example if I record a 17 sec video and record it from the start the first few second is not high quality but I want the video from the start and it's too low quality pls fix that bug ever since you guys add the features it bugs the whole entire platform pls fix it
	Refined	This app is really good, but I have a big problem. When I try to record and edit a video, I want it to be high quality, but it doesn't start off that way. The first few seconds are low quality, and it's been like this since new features were added. Please fix this bug as it affects the whole platform.
	Raw	Good game, last updates were great, but 1 I wish u would consider changing u get a disadvantage the higher ur level 4 instance when ur playing the weekly I'm a 5000 level player & I'm having 2 play with players that could b a thousand level player which is giving them easier boards 2 play, I really hope it's considered 2 change this maybe n the next update, my last review I gave it a 2 star but with the new updates it's a 4 I give it a five if this was changed, thank you & keep up the good work.
	Refined	Good game. The latest updates were great. There is one thing I wish you would consider changing. As you level up, you get a disadvantage. For example, I am a level 5000 player, but I have to play against players who are only a level 1000 player, which gives them easier boards to play. I really hope you will change this in the next update. My last review was a 2-star, but with the new updates, it's now a 4-star. If this issue is fixed, I would give it a 5-star. Thank you and keep up the good work.
Silver Corpus	**Raw**	It's been innumerable number of times where I have booked an Uber the payment mode was upi after dropping me to location and getting the payment done the Uber driver does not verifies the payment and the trip The next time I try to book an Uber it's shows the payment has not been complete and I have to pay the extra amount in my next Trii i have had reported it multiple times drs no action taken from the Uber end It's been more than a month it's still showing 318 payment due it's not a small amount to get it cleared in next trip Second is no Uber or the Uber driver will allow u to leave with our payment after ur trip completion neither driver or Uber is mad such things where people are trying to be looted by Uber should be stopped it does not even makes sense that the customer will leave the desired drop off location without payment I have had also shared my payment screenshot drs no action taken It's disgraceful nd disappointing from a brand like Uber where people use it very frequently nor the customer support is off any use
	Refined	I have had issues with booking an Uber multiple times. The payment mode was incorrect, and the driver didn't verify the payment and trip. The next time I try to book an Uber, it shows the payment hasn't been completed and I have to pay the extra amount in my next trip. I've reported this issue multiple times, but no action has been taken by the Uber team. It's been over a month, and I still have 318 payments due. It's not fair that customers should leave without paying. I also shared my payment screenshot, but no action has been taken. It's disappointing and disappointing from a popular brand like Uber, where people use it frequently. The customer support is not helpful.
	Raw	So i have been using this for over 1 year and i never got too many ads like i do now, at first i used to get 1 ad every hour then i started to get 2 ads every half an hour but then it decrease and i started to get ads after every 2 songs and now i get a add after every song and that too not one ad but 2 ads, tho this app is good and smooth and I understand how thats the only way to get money but i mean 2 ads after each song is just too much
	Refined	I have been using this app for over 1 year. I used to get 1 ad every hour, then 2 ads every half hour, then it decreased to 1 ad every 2 songs. Now, I get 2 ads after every song, not just one ad, but two ads. This app is good and smooth, but I think it's too many ads.
	Raw	Since the new update I have not been able to swipe left and delete shots from watch history like I can do with the videos it is just very inconvenient to delete them individually and sometimes even that doesn't work I have to close the app and open it again and it keeps hanging up like I won't be able to play or something else will not work but the rest will work so I have to close and restart the app it is very frustrating so if you could please just fix this as soon as possible it would be amazing
	Refined	Since the new update, I can't swipe left and delete shots from my watch history like I can do with videos. It's inconvenient to delete them individually. Sometimes, even that doesn't work. I have to close and reopen the app, and it keeps freezing. It's frustrating to have to close and restart the app. Please fix this issue soon.

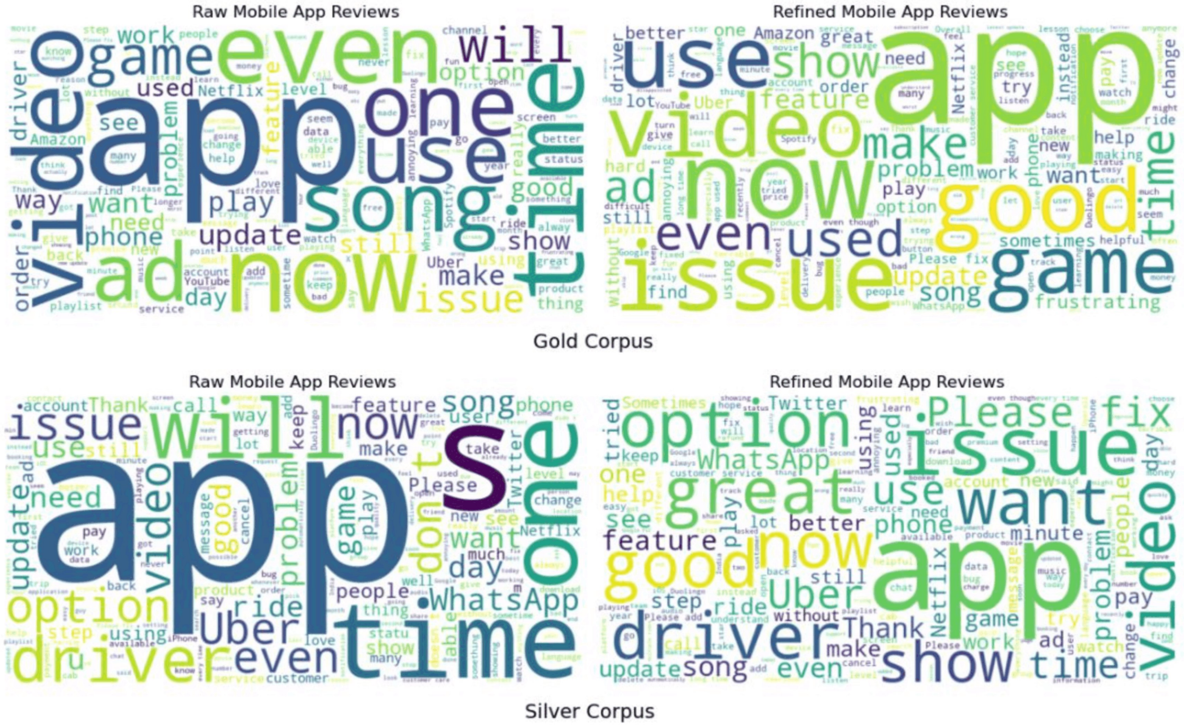
Table 6: Examples of raw and refined reviews from the gold and silver corpus



Word Count Distribution : Gold Corpus

Word Count Distribution : Silver Corpus

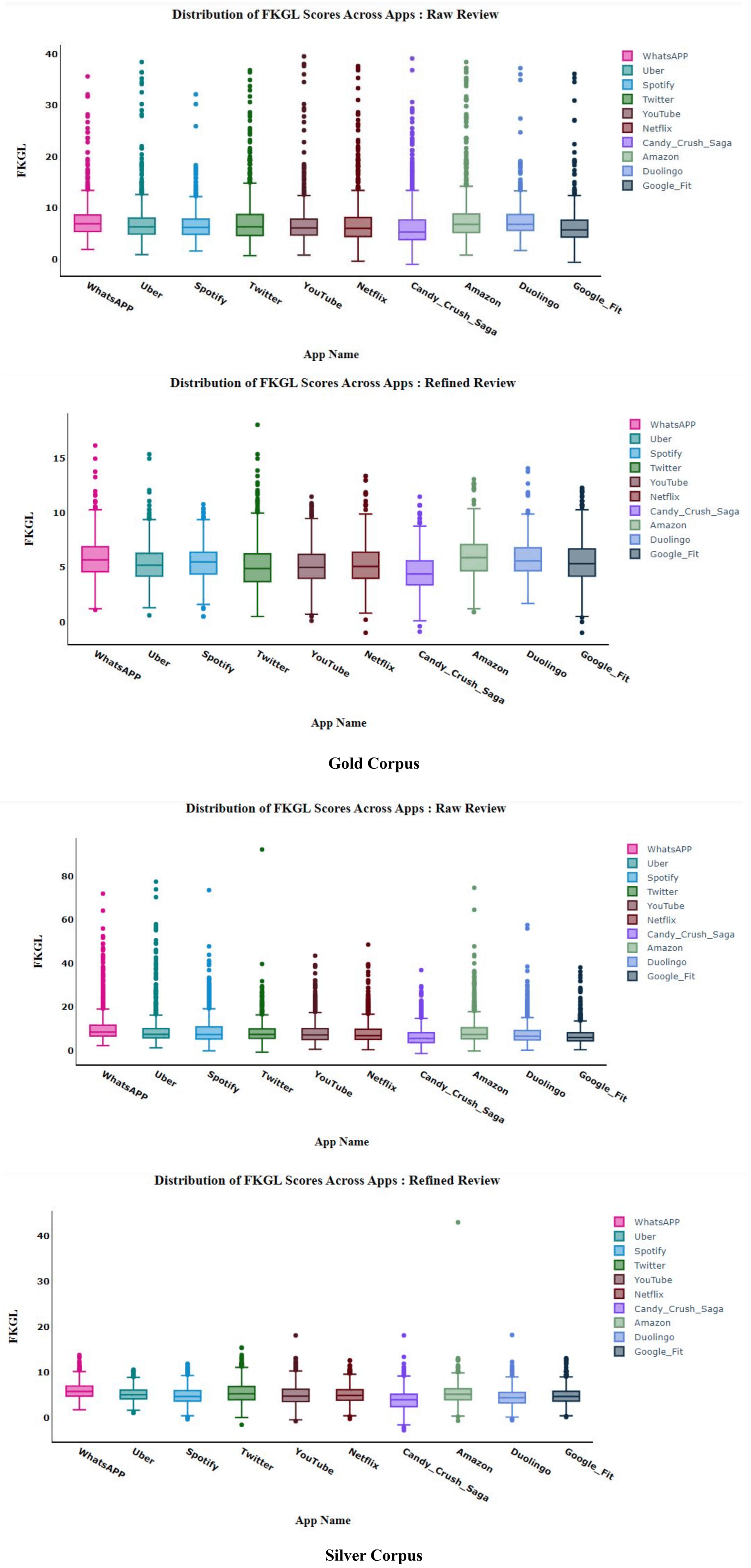
Figure 3: Word Count Distribution of Gold and Silver Corpus Reviews



Gold Corpus

Silver Corpus

Figure 4: Word Cloud of Gold and Silver Corpus Reviews



Distribution of FKGL Scores Across Apps : Raw Review

Distribution of FKGL Scores Across Apps : Refined Review

Gold Corpus

Distribution of FKGL Scores Across Apps : Raw Review

Distribution of FKGL Scores Across Apps : Refined Review

Silver Corpus

Figure 5: FKGL Distribution of Gold and Silver Corpus Reviews