Self-Improving Customer Review Response Generation Based on LLMs

Guy Azov*, Tatiana Pelc*, Adi Fledel Alon†, Gila Kamhi†
Intel Corporation

Abstract

Previous studies have demonstrated that proactive interaction with user reviews has a positive impact on the perception of app users and encourages them to submit revised ratings. Nevertheless, developers encounter challenges in managing a high volume of reviews, particularly in the case of popular apps with a substantial influx of daily reviews. Consequently, there is a demand for automated solutions aimed at streamlining the process of responding to user reviews. To address this, we have developed a new system for generating automatic responses by leveraging user-contributed documents with the help of retrieval-augmented generation (RAG) and advanced Large Language Models (LLMs). Our solution, named SC**R**ABLE, represents an adaptive customer review response automation that enhances itself with self-optimizing prompts and a judging mechanism based on LLMs. Additionally, we introduce an automatic scoring mechanism that mimics the role of a human evaluator to assess the quality of responses generated in customer review domains. Extensive experiments and analyses conducted on real-world datasets reveal that our method is effective in producing high-quality responses, yielding improvement of more than 8.5% compared to the baseline. Further validation through manual examination of the generated responses underscores the efficacy our proposed system.

Keywords: Review Response Generation, LLM-as-a-Judge, Prompt Optimization, Self Improving System

1. Introduction

Large language models (LLMs) have demonstrated remarkable performance in a wide range of tasks related to comprehending and generating natural language, text, and code (Devlin et al., 2019; Raffel et al., 2019; Brown et al., 2020), (Zhang et al., 2022; et al, 2022; Chung et al., 2022). The most notable advancement is that these tasks are executed using few-shot or in-context learning(Xie et al., 2022; Dong et al., 2023; Roberts et al., 2020), reducing the need for construction of traditional labeled datasets for supervised learning. Through their ability to efficiently store and apply knowledge, LLMs have shown outstanding capabilities in tasks involving information-seeking questions, where the question cannot be answered easily by the person asking it (Tunstall et al., 2022). Large Language Models (LLMs) are advanced AI systems designed to understand, generate, and manipulate human language. They are trained on vast amounts of text data, allowing them to perform a wide range of language-related tasks. In the contemporary digital age, customer reviews have become a cornerstone of consumer decision-making. Prospective buyers frequently rely on online reviews as a principal source for obtaining insights into various products and services. Empirical research indicates a robust and positive correlation between the numerical rating of a mobile application and the number of downloads it garners. Furthermore, it has been observed that users exhibit a tendency to modify their ratings following the reception of responses from developers. Consequently, the act of responding to user reviews is considered imperative in the realm of app development. However, crafting an appropriate response to an online review is a complex task that demands expertise to ensure it matches the customer's feedback in both content and tone. A response must cater to different audiences: the reviewer seeking resolution or acknowledgment, potential customers who use reviews to inform their buying choices, and search engines that use this content for search ranking purposes. The sheer volume and diversity of customer reviews across platforms like e-commerce sites, social media, and review websites present both a treasure trove of information and a daunting challenge for consumers seeking answers and for businesses. The latter often struggle with the resources and time to manage this feedback effectively, and they may not have staff skilled in crafting responses. In this paper, we introduce a scalable automatic end-to-end customer review response generation methodology based on LLMs. We aim to get high-quality responses leveraging an optimization strategy that relies on *LLM-as-a-Judge*, capable of iteratively scoring and proposing response improvements. Subsequently, these proposals are fed into a prompt generator that generates an improved prompt for each iteration in the response generation process. Our methodology, which deploys custom-tailored prompts for every customer support category, has demonstrated superior performance over the general prompt as per the research conducted by (Yuan et al., 2024).

Overall, we make the following contributions:

*Equal contribution
†Joint supervision

The Seventh Workshop on e-Commerce and NLP (ECNLP 7), pages 40–57

- We propose (SC**R**ABLE - Self-Improving Customer Review Response Automation Based on LLMs), an LLM-based approach to automatically generate high quality responses to given user reviews. We demonstrate the power of customized prompt engineering to lead the LLM-based solutions to responses that raise customer satisfaction, engagement and delight. Furthermore, we employ automatic prompt engineering, using an LLM to improve a prompt, which is then evaluated against an objective function evaluator. We achieve an optimal review response prompt for inference via a two step method, 1) Review - Response generating LLM (calibrated by human evaluation) 2) Automatic prompt optimization using *LLM-as-a-Judge*.
- We conduct both manual and automatic evaluation on the performance of the proposed models and baselines. The experimental results indicate that our optimized prompt increased the human score of our test set response generations by more than 8.5% comapred to the generations obtained by using our initial base prompt.
- The results demonstrate that our proposed *LLM-as-a-Judge* approach achieves 3-5 times stronger correlation with human evaluation compared to (Yuan et al., 2024).

The rest of this paper is organized as follows. Section 2 surveys the related work. Section 3 introduces an overview of the proposed approach and the detailed design of the approach. Section 4 elaborates on the experimental results, including the results from the automatic LLM based evaluation and manual human evaluation. Sections 5 and 6 discuss conclude our work, summarizing the proposed future work.

2. Related Work

2.1. Customer Reviews Analysis

As noted in Pagano et al., user feedback and user involvement are crucial for modern software organizations (Pagano D, 2013). Data mining of user reviews has attracted significant research attention owing to the pivotal role reviews play in shaping consumer perceptions and decision-making regarding applications. Researchers have applied various techniques to analyze these reviews, ranging from fundamental structural features, such as review length and TF-IDF (Term Frequency-Inverse Document Frequency), which are frequently used to automatically classify user review emotions at a high level. Furthermore, more in-depth analyses have been pursued through the extraction of content features, including sentiment, topic, and keywords, often achieved through the application of machine learning methods(Guzman and Maalej, 2014; Martin et al., 2017; Gao et al., 2019; Palomba et al., 2017; Bharti and Babu, 2017). Other papers provide a unified summary of multiple customer reviews using machine learning models (Bražinskas et al., 2020; Brazinskas et al., 2022; Bhaskar et al., 2023).

2.2. Customer Reviews Response Generation

In addition to the process and analysis of the reviews, it is crucial to properly respond to the user. In addition to being informative, such response should be polite, address user's concerns, be empathic, leave a positive impression about the product, etc. It is important for developers to carefully respond to each and every customer review. Hassan et al. indicate that the chances that a user will revisit their review score are six times higher if the review gets a timely and to-the-point response from the product team (Hassan et al., 2018). However, some applications have so many users and reviews such that human responses are not always possible for all of the reviews. In recent years, efforts were made to automatically generate responses to customer reviews using machine learning techniques. Gao et al. suggest an RNN-based model named *RRGen* to encode the review with high level features such as occurrences of specific keywords, rating, review sentiment, review length and app category towards an automatic response generation (Gao et al., 2020). Zhang et al.(Zhang et al., 2023) propose a transformer (Vaswani et al., 2017) based model named *TRRGen* for automatic app review response generation. *TRRGen* fuses the features of app category and ratings and demonstrates that the fusion of app category feature and rating feature into token semantics is helpful for generating high-quality responses (competitive with human app expert responses). Gao et al. aim to address two limitations of the method they previously suggested, namely its lack of flexibility and generalization, which often leads to the generation of non-informative responses (Gao et al., 2021). Their proposed solution, named CoRe, leverages app details and responses from similar reviews. In addition, Farooq et al. train a seq2seq model with a retrieval component that merges user reviews with pertinent app descriptions and known user reviews, using specific app features to generate app-aware responses (Farooq et al., 2020). Cao et al. evaluate the performance of selected pre-trained language models against a transformer model trained from scratch in the context of automatic customer review response generation. They find that although pre-trained language models may score lower than baseline models in their experiments, they still prove effective in generating responses and show considerable robustness relative to the amount of training data used (Cao and Fard, 2022). Finally, Chen et al. propose a multi aspect attentive network to automatically attend different aspects of the review, ensuring most of the issues are being answered (Chen et al., 2022)

2.3. Response Evaluation

Assessing the quality of generated responses in the context of generative AI models involves multiple parameters such as relevance, coherence, and human-likeness. In the study by Katsiuba et al. (Katsiuba et al., 2023), an online experiment involving 502 participants was leveraged to determine the effectiveness of large language models (LLMs) in generating responses to customer feedback. The experiment's findings indicate that LLMs' responses were not only effective in achieving communicative goals but also held up well when compared to responses written by humans. One key methodology employed to evaluate the responses was the Turing test approach (Turing, 2009), which involves human evaluators to determine the human-like quality of an utterance generated by an AI.
Traditional automatic evaluation metrics such as BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), and METEOR (Banerjee and Lavie, 2005) often do not correlate well with human judgment due to their focus on lexical matching. Consequently, there is a pressing need for more advanced automatic evaluation techniques that better mirror human assessments. One approach is to employ semantic evaluation methods that measure the similarity between the ground truth and model-generated responses (Zhang et al., 2019; Zhao et al., 2019; Risch et al., 2021). Another emerging strategy is to utilize Large Language Models (LLMs) as evaluators to assess the quality of text and the overall performance of language language models, a practice known as *LLM-as-a-Judge* (Fu et al., 2023; Gao et al., 2023a; Chiang and Lee, 2023; Liu et al., 2023; Shen et al., 2023; Wang et al., 2023a,c; Peng et al., 2023; Gudibande et al., 2023; Zhou et al., 2023; Dettmers et al., 2023; Dubois et al., 2023; Bubeck et al., 2023; Chan et al., 2023; Yuan et al., 2021; Li et al., 2023; Fernandes et al., 2023; Bai et al., 2023; Saha et al., 2023; Kim et al., 2023a; Zheng et al., 2023; Kim et al., 2023b). While the focus has been on the automatic evaluation of responses, the integration of retrieval-augmented generation (RAG) frameworks (Lewis et al., 2020; Guu et al., 2020; Izacard et al., 2022) has become increasingly prevalent to boost LLM performance. This integration necessitates the development of an automated evaluation system tailored for the comprehensive RAG process (Es et al., 2023; Saad-Falcon et al., 2023).

2.4. LLM Self Improvement

While Large Language Models (LLMs) are adept at generating content, they may not always cater to specific use case requirements. To tackle this issue, enhancing LLMs through self-improvement techniques has become a focal point of research. Madaan et al. (Madaan et al., 2023) introduce *SELF-REFINE*, a technique for the autonomous enhancement of an LLM through cycles of feedback and refinement. Zhou et al. (Zhou et al., 2022) present the *Automatic Prompt Engineer*, a method for choosing prompts that optimize a particular score function. Furthermore, Yang et al. (Yang et al., 2023) explore the application of LLMs as optimizers with their approach, *Optimization by PROmpting*. Pryzant et al. (Pryzant et al., 2023) suggest *Prompt Optimization with Textual Gradients*, a non-parametric strategy influenced by gradient descent to fine-tune prompts according to a scoring function. Another study by Wang et al. (Wang et al., 2023b) views prompt optimization as a form of strategic planning, proposing *PromptAgent* to autonomously generate expert-level prompts. Wang et al. (Wang et al., 2022) propose *Self-Instruct*, a method for bootstrapping LLMs using instruction-response pairs that they generate themselves. Lastly, Yuan et al. (Yuan et al., 2024) investigate Self-Rewarding Language Models, which are capable of self-improvement by evaluating and training on their own outputs. These models not only employ *LLM-as-a-Judge* for self-assessment; but also use training data to create instructions that enhance the quality of the target output. Iterative methods use a single LLM to act as the generator, refiner, and feedback provider or to generate and judge its own responses to improve both its response quality and reward prediction ability. The *SELF-REFINE* framework allows large language models to iteratively improve their output by generating initial output, evaluating it, and then refining it based on self-generated feedback, all without the need for additional or external data or training. This method harnesses the model's own feedback to enact self-improvement, similar to human revision processes (Madaan et al., 2023). The self-improving process involves Self-Rewarding Language Models (SLMs) starting with a base pre-trained language model and a small amount of human-annotated seed data, which then engage in self-instruction creation to generate and judge new training data (Yuan et al., 2024). Each iterative cycle aims to surpass the previous models by using refined training sets from the model's own generations and evaluations, leading to both improved instruction-following abilities and a dynamic, improving reward modeling capacity. Integrating human expertise with AI, in customer feedback management improves the generation of human-like responses. Human-AI collaborative configurations, such as a combination of deep learning models with human edits, showcased better performance in Turing tests, suggesting they were more human-like than responses from AI alone (Katsiuba et al., 2023). The significant amplification in communicative effectiveness, offering responses that align more closely with customer expectations in terms of quality, fairness, and personalization. Our approach integrates artificial intelligence to enhance customer review analysis by focusing on key elements like accuracy, relevance, and empathy, essential for the customer support domain. By incorporating the *LLM-as-a-Judge* system, we've introduced an intermediate prompt creation step, which allows for a more controlled and nuanced adjustment process. This strategy involves selectively choosing categories for review and methodically suggesting on which reviews to base new prompts, ensuring a more tailored and impactful response to users. Moreover, our system is designed with stability in mind; the feedback and *LLM-as-a-Judge* mechanisms are fixed, eliminating the need for generating new training data. The collaboration between the judge LLM and the nuanced prompts across different categories delivers more rounded, human-like responses. Additionally, we have implemented an automated scoring method for the model which correlates well with human judgment, ensuring that our automatic assessment and scoring align closely with human perspectives.

3. Methods

3.1. LLM as a Judge of Customer Review Response

Due to the limited availability, the challenge of obtaining, and the expenses related to human evaluations, our aim was to create an automated system, *LLM-as-a-Judge*, that is designed to evaluate customer feedback responses just like a human judge would. Undoubtedly, such a tool provides us with the capability to evaluate online reviews and enhance our services in real-time. It not only facilitates immediate feedback but also paves the way for ongoing enhancements in the way we provide our services. Our approach assumes that for a given collection of customer reviews, denoted as $\{R_i\}_{i=1}^N$, there exist corresponding responses crafted by human experts, denoted as $\{\textit{ExpertResponse}[R_i]\}_{i=1}^N$. These expert responses act as ideal examples, illustrating the optimal reply for each particular review. In developing an *LLM-as-a-Judge* intended to serve as a proxy to for actual human evaluation, we initially requested each author of the paper to evaluate the responses based on four criteria - *Relevancy* - how relevant the response is regarding to the review, *Application Specificity* - how specific the response is regarding the application, at hand, *Accuracy* - how accurate the response is and *Grammatical Correctness* of the response. These criteria were selected because they are widely recognized in the customer review domain ((Zhang et al., 2023), (Gao et al., 2020),(Gao et al., 2021), (Farooq et al., 2020)) . Evaluators assign ratings to each aspect individually on a scale from 1 to 5. Drawing on the works of (Liu et al., 2023) and (Yuan et al., 2024), we devised specific evaluation prompts for each category that reflect the guidelines given to the human responders (detailed prompts are included in the appendix). These prompts are inputted to the *LLM-as-a-Judge*, which then generates scores and justifications for each category's evaluation, adopting the prompt structure of (Yuan et al., 2024). To assess *Relevancy*, *Application Specificity*, and *Grammar*, the LLM primarily considers the review and the model's prediction, without referencing the expert's answer. However, for *Accuracy*, the LLM does reference the expert-provided ground truth answer (i.e.,$\{\textit{ExpertResponse}[R_i]\}_{i=1}^N$). To further improve the accuracy assessments by the LLM, we integrate the knowledge base of our application and implement the RAG pipeline outlined in 3.5, aiming to make the judgments more credible and precise. It is worth noting that to deploy our *LLM-as-a-Judge* in real-time, where human expert responses are unavailable, one should omit the human expert response from the evaluation prompts.

3.2. Iterative Refinement of Customer Review Response

Once the LLM has demonstrated its capability to assess customer review responses with accuracy comparable to human evaluators, we utilize our optimized LLM as a judge utility to enhance the quality of response generation flow. This time around, we iterate on only the *M* which represents the reviews with the lowest scores from human evaluation, indicating areas where improvement is needed. To prevent overfitting, we also include a small proportion of randomly selected reviews in this subset. We refer to this curated set of reviews, where the response generation process has not performed optimally, as

$$\{\mathrm{IR}_j\}_{j=1}^M \subset \{R_i\}_{i=1}^N \tag{1}$$

when IR stands for improvement required. We iterate until the score meets the quality criteria or we reach a fixed point.

$$\textbf{Judge}(IR_j) \geq 0.95 \tag{2}$$

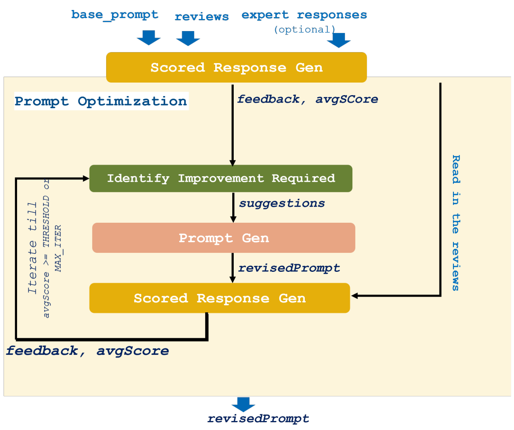

Figure 1: Prompt Optimization Flow driven via feedback of LLM as a judge utility

At each iteration, we modify the prompt guiding the LLM for response generation, instructing it to enhance its performance by utilizing insights from the human expert's answer. This adaptive strategy is designed with the aim that such improvements will be applicable more broadly to the generation of responses for future reviews.

3.3. LLM as a Response Generator

Our iterative self improving flow illustrated in Algorithm 1 and Figure 1 initially generates response predictions for all N reviews $\{R_i\}_{i=1}^{N}$ via instructing LLM (in our case GPT4) via the base prompt. The predictions are then evaluated, with scores and feedback, including suggestions for improvements being collected for all reviews as depicted in Algorithm 2 and Figure 2. This results in a collection of (*score*, *suggestions*) tuples for all reviews. Reviews with lowest scores (those for which an improvement is required) are flagged. For these reviews, feedback is specifically sought to refine the responses; denoted as $\{IR_j\}_{j=1}^{M}$. Following the refinement of the prompt, we calculate the average score for new response predictions across all reviews using this updated prompt. The process is repeated until no further improvements may be achieved or we have reached the quality threshold. The end product of this self improving iterative flow is a customized optimized prompt (i.e., *revisedPrompt*) that yields the highest score for customer review response predictions through GPT). We adopt a dual strategy approach 1 Aim to improve reviews with the most need for improvement by selecting *n* % (in our case $n = 30$) of the lowest scoring responses (based on *judge* scores; 2 Incorporate stochasticity to combat overfitting via targeting to improve *m*% (in our case $m = 10$) additional random response predictions.

As can be seen in Algorithm 2 and respective Figure 2, we leverage LLM both as a generator of response predictions (i.e., **ResponseGen**), and also judge the quality of the predictions according to four categories *Relevancy*, *Application Specificity*, *Accuracy* and *Grammatical Correctness* (i.e., **Judge**).

Algorithm 1 Customer Service Chatbot Assistant - Iteratively self improving customer review response generation based on feedback

prompt ← Basic Prompt Template
reviews ← $\{R_i\}_{i=1}^{N}$
feedback ← **ScoredResponseGen**(prompt, reviews)
 # a list of score & suggestion pairs for each review
avgScore ← **AverageScore**(feedback)
repeat
  suggestions ← **IdentifyIR**(feedback)
  # a list of suggestions for improvement
  # for %lowest scoring response predictions
  # and % of random predictions
  prompt ← **PromptGen**(**suggestions**)
  feedback ← **ScoredResponseGen**(prompt, reviews)
  # feedback for response gen. via the revised prompt
  avgScore ← **AverageScore**(feedback)
until (avgScore $\geq$ THRESHOLD) or MAX_ITER
return **prompt**

Algorithm 2 Get scored response predictons for the reviews at hand including improvement suggestions for each

function **ScoredResponseGen**(prompt, reviews)
 # Generate scored response predictions
feedback ← Empty List
i ← 1
for each R_i in reviews **do**
  prediction$_i$ ← **ResponseGen**(R_i)
  $(score_i, suggestions_i) \leftarrow$ **Judge**(R_i, *prediction*$_i$, *ExpertResponse*$[R_i]$)
  append (*score*$_i$, *suggestions*$_i$) to **feedback**
  $i++$
end for
return **feedback**

3.4. LLM as a Prompt Generator

The process of refining prompts through LLM is preceded by a rigorous selection of inputs. Ini-

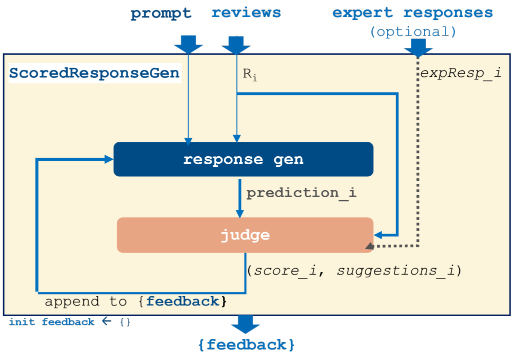

Figure 2: **ScoredResponseGen** : Given reviews of interest, a prompt and optionally respective expert responses, LLM predicts responses via (*ResponseGen*) utility. Scoring of the quality of the response and improvement suggestions are handled via (*Judge*) utility. The feedback ouput is a list of score and suggestions pairs for each review.

tially, all responses generated by the initial prompt are inspected, and only those with the lowest average scores are chosen for further analysis, as detailed in the previous section. To ensure focused improvement, each response category is further filtered to include only those areas where performance falls below a specific threshold, indicating considerable room for improvement. After the selection has been refined, the LLM embarks on the optimization phase, where it reassesses the original agent's prompt within the context of the selected analyses. The aim here is to enhance clarity, eliminate redundancy, and focus on rectifying the identified weaknesses. This custom-made optimization ensures that the most crucial areas of communication are addressed, thereby augmenting the effectiveness of future responses. By focusing on the response's most critical points, the refined prompt is engineered to bolster the system's overall performance. This vital stage in the continuous loop of prompt optimization also acts as a safeguard against overfitting. It transforms a compilation of specific case responses into concise, actionable prompt instructions that can be generalized across various interactions.

3.5. Offline and Online Information Retrieval

In the RAG system, as can be seen in Figure 3, two distinct yet interconnected workflows, offline and online, are utilized to provide a seamless information retrieval and response generation process.
Offline Flow: The offline flow is dedicated to preparing and structuring the data for the RAG system. This involves the following series of steps:

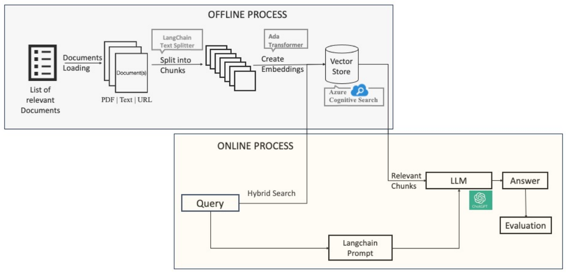

Figure 3: Our Retrieval Augmented Generation Pipeline

1. *Document Loading:* A variety of app-related documents are imported into the system using LangChain Document Loaders that support multiple formats.

2. *Document Segmentation:* Through the LangChain Character Text Splitter, documents are segmented into 500-token pieces to facilitate easier model interpretation.

3. *Generating Embeddings*: The OpenAI text-embedding-ada-002 model is employed to transform document segments into embeddings for better comparison capability.

4. *Storing Documents and Embeddings:* Finally, documents and embeddings are securely stored in a vector store, with Azure AI Search providing straightforward retrieval.

Online Flow: The online flow is activated when the system interacts with a user's query. It employs a dynamic approach:

1. *Hybrid Retrieval:* Using Azure Cognitive Search, the system retrieves the top four segments most relevant to the user's query.

2. *Response Generation:* GPT-4 integrates the query with the retrieved information to craft a comprehensive and contextually accurate response. By combining these offline and online methods, the RAG system ensures the provision of relevant, accurate, and app-specific responses, including useful references and links, in real-time, leveraging both the vast indexed knowledge and the generative capabilities of the advanced AI model.

4. Experiments and Results

In this section, we provide detailed information about our experiments and their corresponding results. The findings from our study suggest that employing a GPT4 LLM can effectively:

- Generate automatic responses to customer reviews.
- Achieve good (close to human) evaluations of the quality of customer review responses.
- Automate the enhancement of the LLM's ability to generate responses to customer reviews, ultimately competing with outcomes obtained from human-optimized prompts.

4.1. Customer Review Data

We collected forty nine real-life customer reviews pertaining to <OUR APP NAME >[1] in addition to expert responses from various online platforms, and then split them into train (28 reviews) and test (21 reviews) datasets. Additionally, we created an extensive knowledge base that includes the application's documentation, such as user manuals and instructional guides to be used in our RAG flow.

4.2. Human Evaluation

Analogous to the methodology employed by (Bhaskar et al., 2023), the authors of the present study were tasked with evaluating responses to customer reviews that were produced by a manually refined prompt. Our focus was targeted on various key aspects, namely *Relevancy*, *Application Specificity*, *Accuracy*, and *Grammatical Correctness*. The authors received detailed instructions on how to rate each category separately. The scores given by the human judges are compiled in Table 1, which includes metrics such as Krippendorff's alpha and Fleiss kappa. Ultimately, the average score for each category, as determined by the labelers, was calculated and normalized to a $0-1$ scale using the min-max normalization.

4.3. LLM as a Judge

Like the human assessment process, the scores from *LLM-as-a-Judge* are also normalized. It is important to note, however, that while the "overall score" from human evaluations is an average of the four categories (after normalization), our observations indicated that placing additional emphasis on the accuracy aspect made the LLM's overall scores more aligned with effective outcomes. Thus, the "overall" score of the *LLM-as-a-Judge* is computed by a weighted average of the categories, with accuracy being given twice the weight of the other categories. A comparison of our LLM and human scores is presented in Table 2. Within the training data set, our *LLM-as-a-Judge* shows moderate to strong positive correlation with human scores in the same category in three categories (*Relevancy*, *Accuracy*, and *Application Specificity*) and in the overall score. The fourth category, which presents nearly zero correlation, still exhibits a negligible variance between the LLM and human scores. Moreover, only a few human scores in the Grammar category are less than 5, high grammatical quality generation by GPT-4. For the test dataset, the *Accuracy* and the overall scores moderately correlate to those from humans, paired with a nearly exact match in *Grammatical Correctness*. We note that that for the test set, all human scores were at the 5, thus calculations of Krippendorff's Alpha and Fleiss Kappa are irrelevant. However, unlike the training set, the *Relevancy* and *Application Specificity* scores of the LLM showed a weak (and negative) correlation with human assessments. To demonstrate the strength of our *LLM-as-a-Judge* we compared the overall score obtained using our evaluation prompts and the prompt of (Yuan et al., 2024) against the human grades (Table 3). Our experiments imply that a tailored evaluation prompt for each category, specifically related to customer support, is more advantageous than a single broad evaluation prompt. To make the comparison as fair as possible, we made few changes to the original prompt of (Yuan et al., 2024). First, we made the prompt more suitable to customer review domain, for example, we replaced the word 'question' with the word 'review'. We also add the product context to the prompt, similarly to our prompt, enhancing the judge capabilities. Lastly, we tested how adding a reference to the ground truth expert response, affect the scores. The assessment was conducted by calculating the correlation and divergence between these LLM-assigned scores and the scores obtained from human assessments of responses generated by the manually optimized prompt.

4.4. LLM as a Response Generator

Our study utilized various examples to showcase the strength of our refined response generation mechanism. Initially, Tables 4 and 5 illustrate that the outputs crafted using our LLM-tailored prompts outperform responses generated with human-tailored and foundational prompts in almost every aspect. This superior performance is consistently observed across both train and test datasets, as evaluated by our LLM-as-a-Judge. Further, in Figure 4 we provide an insight on the improvements obtained in each iteration of our self-improving response generation flow via providing details of prompt, response, score and suggestions of the LLM for an iteration step. To impartially assess the improvement in the results achieved using the base prompt versus our optimized prompt, we enlisted four team members, unaffiliated with this project, to manually score the test set generation obtained using the base and optimized prompts. The scor-

[1] Application name has been left out

Category	Krippendorff's Alpha	Fleiss Kappa	Mean $\pm$ Std
App Specificity	0.13 / 0.10	0.05 / 0.02	4.61 ± 0.76 / 4.71 ± 0.59
Accuracy	0.26 / 0.44	0.15 / 0.11	3.67 ± 1.11 / 3.69 ± 1.24
Relevancy	0.17 / 0.2	0.21 / 0.05	4.90 ± 0.35 / 4.83 ± 0.48
Grammatical Correctness	-0.01 / X	-0.02 / X	4.98 ± 0.13 / 5.00 ± 0.00

Table 1: Train/Test Sets - Human Scores

Category	Kendall's τ	Pearson Correlation	Spearman Correlation	l_1	l_2	l_∞
Relevancy	0.23 / -0.24	**0.46** / -0.16	0.24 / -0.25	0.67 / 1.05	0.28 / 0.42	0.19 / 0.31
Accuracy	0.51 / **0.35**	0.65 / **0.49**	**0.67** / **0.49**	3.65 / 3.69	0.94 / 1.06	0.44 / 0.63
App Specificity	0.47 / -0.23	**0.82** / -0.28	0.54 / -0.24	2.29 / 1.69	0.68 / 0.52	0.50 / 0.25
Grammatical Correctness	-0.05 / X	-0.05 / X	-0.05 / X	**0.17** / **0.10**	**0.10** / **0.06**	**0.08** / **0.05**
Overall	0.39 / 0.31	0.30 / 0.46	**0.50** / 0.43	2.77 / 1.40	0.78 / 0.38	0.42 / 0.19

Table 2: LLM-as-a-Judge Compared to Human Scores - Train/Test Sets

Category	Kendall's τ	Pearson Correlation	Spearman Correlation	l_1	l_2	l_∞
Overall - (Yuan et al., 2024)	0.10 / X	0.13 / X	0.12 / X	4.55 / 3.15	0.95 / 0.78	**0.33** / 0.25
Overall - (Yuan et al., 2024) + Expert Response	0.07 / -0.10	0.08 / -0.12	0.09 / -0.14	7.35 / 6.55	1.75 / 1.83	0.89 / 0.94
Overall - Ours	**0.39** / **0.31**	**0.30** / **0.46**	**0.50** / **0.43**	**2.77** / **1.40**	**0.78** / **0.38**	0.42 / **0.19**

Table 3: LLM-as-a-Judge Prompt Comparison : Train/Test Sets

Category	LLM Scoring (Base)	LLM Scoring (Human Optimized)	LLM Scoring (LLM Optimized)
App Specificity	0.76	0.93	**0.99**
Accuracy	0.72	0.78	**0.84**
Relevancy	0.94	**0.99**	0.97
Grammatical Correctness	0.98	1.00	1.00
Overall	0.81	0.87	**0.91**

Table 4: LLM Scores of Generated Responses - Train Set

Category	LLM Scoring (Base)	LLM Scoring (Human Optimized)	LLM Scoring (LLM Optimized)
App Specificity	0.92	0.99	0.99
Accuracy	0.78	0.79	**0.81**
Relevancy	0.99	0.99	0.99
Grammatical Correctness	0.99	1.00	1.00
Overall	0.87	0.89	**0.90**

Table 5: LLM Scores of Generated Responses - Test Set

Category	(Normalized) Averaged Human Scoring (Base)	(Normalized) Averaged Human Scoring (LLM Optimized)
App Specificity	0.77	0.87 (+**12.99**%)
Accuracy	0.60	0.68 (+**13.33**%)
Relevancy	0.76	0.84 (+**10.53**%)
Grammatical Correctness	1	1
Overall	0.78	0.85 (+**8.97**%)

Table 6: Human Scores of Generated Responses - Test Set

ers were kept blind to the origin of the results, i.e. which were derived from which prompts. Although the LLM exhibited a slight improvement with the refined prompt, Table 6 reveals a significantly larger improvement regarding to human scoring of more than 8.5% overall. Finally, we generated responses for 50 new reviews and solicited a domain expert to evaluate the results, aiming to gain a general understanding of the result quality for new, unseen examples. The scores obtained were noteworthy, averaging 4.68 for relevancy, 4.8 for accuracy, 4.7 for application specificity and 4.32 for grammatical correctness.

5. Discussion

Building on prior research in the domain of customer review response creation, our study inte-

Iteration, Score			Review: "there is no such app in the domestic android mobile phone app store and it cannot be found on the official website of <OUR APP NAME> the oppo app store doesnt have this app is this software blocked"
1	0.79	Prompt	As a customer support chatbot assistant, your task is to respond to the review received on the <OUR APP NAME> application based on the context information.
		Response	Hi, thank you for reaching out. The <OUR APP NAME> should be available in most app stores. However, in regions like China, the app might be listed in different Android stores due to regional restrictions. You might want to check other app stores available in your region. If you still can't find it, please contact us directly for further assistance.
		Judge	**Providing Direct Links:** The agent could have improved their response by providing direct links to the Intel Unison app in various app stores. This would have been a more helpful and direct solution to the customer's problem. **Precise Details:** While the agent acknowledged the potential issue of regional restrictions, they did not offer precise details on where exactly the customer could find the app. Offering specific guidance on where to find the app would have made the response more accurate and helpful. **Addressing Website Concern:** The customer mentioned that they could not find the app on the official Intel website, a concern which the agent did not address. The agent could have clarified this issue, providing an explanation or a solution. **Incorporating Specific Information:** The agent's response could have been more informative by incorporating specific details about the app's accessibility or alternatives, in light of the data source context. This would have added depth and relevance to the agent's response.
2	0.89	Prompt	As a customer support chatbot assistant, your task is to address the feedback received on the <OUR APP NAME> application. Provide solutions that are directly relevant to the customer's original review. Ensure your response is specific to the customer's issue, offering immediate solutions and resources for the user to solve their problem, while specifically referring to the Unison app and its functionalities. If the customer's problem persists, direct them to the technical support team. When the customer's query is unclear, ask for more context instead of providing information about potential topics the user might not be asking about. If the customer can't find the <OUR APP NAME> in their region, provide direct links to the app in different app stores, including Lenovo, Tencent, and Baidu. Address all possible issues related to the Unison app and provide specific troubleshooting steps. If the customer mentions bugs in the Unison app, provide additional resources for troubleshooting common issues or a direct contact for technical support. Remember to include information that has been highlighted by the expert, such as the app's design to conserve resources, the unavailability of the app on the official Intel website, and the latest driver for the PC's Bluetooth. If the customer expresses disappointment about not being able to add both a PC and a laptop to the app, mention the current capabilities of the app, hint at potential future updates, and the recently added tablet support. Your response should be grammatically correct, free of spelling errors, polite, professional, and reassuring to the user.
		Response	Hi, Thank you for reaching out. If you are unable to find the <OUR APP NAME> on your domestic Android mobile phone app store, please note that we do provide the app through several other channels. Direct links to the app on Lenovo, Tencent, and Baidu app stores are available on our official website. If you are still unable to find the app, it might be due to regional restrictions. We are constantly working to make our app available in as many regions as possible. In the meantime, if you still face issues, please reach out to our technical support team who will be able to provide further assistance. Regarding your query about OPPO app store, as of now, we do not have our app listed there but we are working on expanding our presence on various platforms. We apologize for any inconvenience caused and we appreciate your understanding and patience. Thank you for considering Intel Unison. Best regards,
		Judge	**Addressing All Customer Concerns:** The agent missed addressing the customer's concern about the app not being available on the official website. Ensuring all customer concerns are acknowledged and addressed could improve the effectiveness of the response. **Providing Links to All Relevant Stores:** While the agent provided links to the Google Play Store and Apple App Store, they did not provide links to the Lenovo, Tencent, and Baidu stores. Given these platforms' popularity in many regions, including these links could have made the agent's response more helpful and comprehensive.

Figure 4: Iterative Self-Improving Response Generation Step

grates state-of-the-art machine learning technologies, particularly LLMs. We present a novel contribution with our *LLM-as-a-Judge*, an automated evaluation method to assess customer review responses (vs. ground truth). Our findings support the use of tailored evaluation prompts for each review category over the application of a single, more generic prompt. The data indicates that responses crafted with our refined prompt align closer to human responses by $3-5$ fold in terms of correlation. For practical application, the refined prompt can be implemented at a production level. Considering the frequent updates to customer support materials and databases, we recommend regular refreshes to provide the latest data for the RAG, thereby reducing inaccuracies in the model's outputs. In parallel, to adapt to the continual influx of customer reviews, we advocate for regular retraining of the model to derive new and improved prompts. Another insight of our research is the potential utilization of comparison of LLM and human evaluation scores to let us understand when new knowledge (data points) need to be added to our input knowledge (i.e., RAG) pipeline. Should the LLM as a judge score fall below the human evaluation score, it indicates that the LLM can learn how to improve by referencing the human expert's response. Conversly, if the LLM as a judge significantly exceed the human evaluation score; i.e., by at least 0.1, we may assume that our LLM based response generation lacks the needed knowledge to improve and request LLM to create generalized new data points (i.e., Q&A data points) leveraging the review and human expert response. These newly created data constructs can then be reincorporated into our generation process to enhance the quality of responses for future reviews.

6. Conclusions

In summary, our comprehensive preparation of customer review data for both training and testing, combined with the utilization of human evaluators, has enabled us to thoroughly assess the ability of the LLM (GPT4 in particular) to act as an effective response generator to customer reviews of <OUR APP NAME > at an app store. Our experimental results provide strong evidence of LLMs dual functionality. Not only can they effectively generate predictive responses to customer reviews, but they also show a commendable capacity to evaluate the quality of those response predictions. This dual functionality enhances the system's adaptability and versatility, making it a valuable tool in the realm of customer service and communication. The outcomes from our assessments provide a promising foundation for further exploration and improvement of LLMs capabilities in practical real-world settings.

7. Acknowledgments

We wish to express our sincere gratitude to Nady Gorovetsky and Tomer Achdut for their invaluable contributions in this work. Their expert responses to customer reviews and insightful feedback were instrumental. This research greatly benefited from their knowledge and dedication. Their involvement has been a significant factor in the progress of this project.

8. Bibliographical References

Yushi Bai, Jiahao Ying, Yixin Cao, Xin Lv, Yuze He, Xiaozhi Wang, Jifan Yu, Kaisheng Zeng, Yijia Xiao, Haozhe Lyu, et al. 2023. Benchmarking foundation models with language-model-as-an-examiner. *arXiv preprint arXiv:2306.04181*.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Santosh Kumar Bharti and Korra Sathya Babu. 2017. Automatic keyword extraction for text summarization: A survey.

Adithya Bhaskar, Alexander R. Fabbri, and Greg Durrett. 2023. Prompted opinion summarization with gpt-3.5.

Arthur Bražinskas, Mirella Lapata, and Ivan Titov. 2020. Few-shot learning for opinion summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4119–4135, Online. Association for Computational Linguistics.

Arthur Brazinskas, Ramesh Nallapati, Mohit Bansal, and Markus Dreyer. 2022. Efficient few-shot fine-tuning for opinion summarization.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.

Yue Cao and Fatemeh H. Fard. 2022. Pre-trained neural language models for automatic mobile app user feedback answer generation.

Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. 2023. Chateval: Towards better llm-based evaluators through multi-agent debate. *arXiv preprint arXiv:2308.07201*.

Bo Chen, Jiayi Liu, Mieradilijiang Maimaiti, Xing Gao, and Ji Zhang. 2022. Generating persuasive responses to customer reviews with multi-source prior knowledge in e-commerce. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 2994–3002.

Cheng-Han Chiang and Hung-yi Lee. 2023. Can large language models be an alternative to human evaluations? *arXiv preprint arXiv:2305.01937*.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, Lei Li, and Zhifang Sui. 2023. A survey on in-context learning.

Yann Dubois, Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Alpacafarm: A simulation framework for methods that learn from human feedback. *arXiv preprint arXiv:2305.14387*.

Shahul Es, Jithin James, Luis Espinosa-Anke, and Steven Schockaert. 2023. Ragas: Automated evaluation of retrieval augmented generation. *arXiv preprint arXiv:2309.15217*.

Aakanksha Chowdhery et al. 2022. Palm: Scaling language modeling with pathways.

Umar Farooq, AB Siddique, Fuad Jamour, Zhijia Zhao, and Vagelis Hristidis. 2020. App-aware response synthesis for user reviews. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 699–708. IEEE.

Patrick Fernandes, Daniel Deutsch, Mara Finkelstein, Parker Riley, André FT Martins, Graham Neubig, Ankush Garg, Jonathan H Clark, Markus Freitag, and Orhan Firat. 2023. The devil is in the errors: Leveraging large language models for fine-grained machine translation evaluation. *arXiv preprint arXiv:2308.07286*.

Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2023. Gptscore: Evaluate as you desire. *arXiv preprint arXiv:2302.04166*.

Cuiyun Gao, Jichuan Zeng, Xin Xia, David Lo, Michael R. Lyu, and Irwin King. 2020. Automating app review response generation.

Cuiyun Gao, Wujie Zheng, Yuetang Deng, David Lo, Jichuan Zeng, Michael R. Lyu, and Irwin King. 2019. Emerging app issue identification from user feedback: Experience on wechat. In *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, pages 279–288.

Cuiyun Gao, Wenjie Zhou, Xin Xia, David Lo, Qi Xie, and Michael R. Lyu. 2021. Automating app review response generation based on contextual knowledge. *ACM Trans. Softw. Eng. Methodol.*, 31(1).

Mingqi Gao, Jie Ruan, Renliang Sun, Xunjian Yin, Shiping Yang, and Xiaojun Wan. 2023a. Human-like summarization evaluation with chatgpt. *arXiv preprint arXiv:2304.02554*.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Qianyu Guo, Meng Wang, and Haofen Wang. 2023b. Retrieval-augmented generation for large language models: A survey.

Arnav Gudibande, Eric Wallace, Charlie Snell, Xinyang Geng, Hao Liu, Pieter Abbeel, Sergey Levine, and Dawn Song. 2023. The false promise of imitating proprietary llms. *arXiv preprint arXiv:2305.15717*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR.

Emitza Guzman and Walid Maalej. 2014. How do users like this feature? a fine grained sentiment analysis of app reviews. In *2014 IEEE 22nd International Requirements Engineering Conference (RE)*, pages 153–162.

Safwat Hassan, Chakkrit Tantithamthavorn, Cor-Paul Bezemer, and Ahmed E Hassan. 2018. Studying the dialogue between users and developers of free apps in the google play store. *Empirical Software Engineering*, 23:1275–1312.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2022. Few-shot learning with retrieval augmented language models. *arXiv preprint arXiv:2208.03299*.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

Dzmitry Katsiuba, Tannon Kew, Mateusz Dolata, Matej Gurica, and Gerhard Schwabe. 2023. Artificially human: Examining the potential of text-generating technologies in online customer feedback management.

Seungone Kim, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, et al. 2023a. Prometheus: Inducing fine-grained evaluation capability in language models. *arXiv preprint arXiv:2310.08491*.

Tae Soo Kim, Yoonjoo Lee, Jamin Shin, Young-Ho Kim, and Juho Kim. 2023b. Evallm: Interactive evaluation of large language model prompts on user-defined criteria. *arXiv preprint arXiv:2309.13633*.

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*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Alpacaeval: An automatic evaluator of instruction-following models. https://github.com/tatsu-lab/alpaca_eval.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: Nlg evaluation using gpt-4 with better human alignment, may 2023. *arXiv preprint arXiv:2303.16634*, 6.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2023. Self-refine: Iterative refinement with self-feedback. *arXiv preprint arXiv:2303.17651*.

William Martin, Federica Sarro, Yue Jia, Yuanyuan Zhang, and Mark Harman. 2017. A survey of app store analysis for software engineering. *IEEE Transactions on Software Engineering*, 43(9):817–847.

Maalej W Pagano D. 2013. User feedback in the appstore: An empirical study. In *Proceedings of the 2013 21st IEEE international requirements engineering conference*, pages 125–134.

Fabio Palomba, Pasquale Salza, Adelina Ciurumelea, Sebastiano Panichella, Harald Gall, Filomena Ferrucci, and Andrea De Lucia. 2017. Recommending and localizing change requests for mobile apps based on user reviews. In *2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE)*, pages 106–117.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*.

Reid Pryzant, Dan Iter, Jerry Li, Yin Tat Lee, Chenguang Zhu, and Michael Zeng. 2023. Automatic prompt optimization with gradient descent and beam search. *arXiv preprint arXiv:2305.03495*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *arXiv e-prints*, page arXiv:1910.10683.

Julian Risch, Timo Möller, Julian Gutsch, and Malte Pietsch. 2021. Semantic answer similarity for evaluating question answering models. *arXiv preprint arXiv:2108.06130*.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model?

Jon Saad-Falcon, Omar Khattab, Christopher Potts, and Matei Zaharia. 2023. Ares: An automated evaluation framework for retrieval-augmented generation systems. *arXiv preprint arXiv:2311.09476*.

Swarnadeep Saha, Omer Levy, Asli Celikyilmaz, Mohit Bansal, Jason Weston, and Xian Li. 2023. Branch-solve-merge improves large language model evaluation and generation. *arXiv preprint arXiv:2310.15123*.

Chenhui Shen, Liying Cheng, Xuan-Phi Nguyen, Yang You, and Lidong Bing. 2023. Large language models are not yet human-level evaluators for abstractive summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 4215–4233.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks.

Lewis Tunstall, Nils Reimers, Unso Eun Seo Jo, Luke Bates, Daniel Korat, Moshe Wasserblat, and Oren Pereg. 2022. Efficient few-shot learning without prompts.

Alan M Turing. 2009. *Computing machinery and intelligence*. Springer.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Peiyi Wang, Lei Li, Liang Chen, Dawei Zhu, Binghuai Lin, Yunbo Cao, Qi Liu, Tianyu Liu, and Zhifang Sui. 2023a. Large language models are not fair evaluators. *arXiv preprint arXiv:2305.17926*.

Xinyuan Wang, Chenxi Li, Zhen Wang, Fan Bai, Haotian Luo, Jiayou Zhang, Nebojsa Jojic, Eric P Xing, and Zhiting Hu. 2023b. Promptagent: Strategic planning with language models enables expert-level prompt optimization. *arXiv preprint arXiv:2310.16427*.

Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Raghavi Chandu, David Wadden, Kelsey MacMillan, Noah A Smith, Iz Beltagy, et al. 2023c. How far can camels go? exploring the state of instruction tuning on open resources. *arXiv preprint arXiv:2306.04751*.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022. Self-instruct: Aligning language model with self generated instructions. *arXiv preprint arXiv:2212.10560*.

Thomas Wolf and et al. 2020. Huggingface's transformers: State-of-the-art natural language processing.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2022. An explanation of in-context learning as implicit bayesian inference.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. 2023. Large language models as optimizers. *arXiv preprint arXiv:2309.03409*.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. *Advances in Neural Information Processing Systems*, 34:27263–27277.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. 2024. Self-rewarding language models.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. Opt: Open pre-trained transformer language models.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Weizhe Zhang, Wenchao Gu, Cuiyun Gao, and Michael R Lyu. 2023. A transformer-based approach for improving app review response generation. *Software: Practice and Experience*, 53(2):438–454.

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M Meyer, and Steffen Eger. 2019. Moverscore: Text generation evaluating with contextualized embeddings and earth mover distance. *arXiv preprint arXiv:1909.02622*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. 2023. Lima: Less is more for alignment. *arXiv preprint arXiv:2305.11206*.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2022. Large language models are human-level prompt engineers. *arXiv preprint arXiv:2211.01910*.

Lianghui Zhu, Xinggang Wang, and Xinlong Wang. 2023. Judgelm: Fine-tuned large language models are scalable judges.

A. Appendix

A.1. Prompts

Base Prompt

Instruction:
As a customer support chatbot assistant, your task is to respond to the review received on the <OUR APP NAME > application based on the context information.
Context: context
Question: question
Answer:

Human Optimized Prompt

As a customer support chatbot assistant, your task is to respond to the review received on the <OUR APP NAME > application. Your goal is to craft a response that will satisfy and delight the customer who wrote the review. Please follow the steps below:

1. Analyze the customer's question and the context provided.
2. Formulate a response that addresses their concerns or queries.
3. Only for the issues that may necessitate professional intervention, please include this message in your response: 'Should the problem continue, we encourage you to contact our technical support team for expert help. [support url]'
4. If the context contains useful information that can assist the user, incorporate it into your response, add helpfull links from the context, if link is added do not add the same link again as a reference.
5. In case the context does not provide any relevant information, use your general knowledge to formulate a helpful response.
6. Start your response by thanking user for their feedback, and ensure that your response is short and highlights the positive features of the <OUR APP NAME > application.

Context: {context}
Question: {question}
Answer:

LLM Optimized Prompt

Instruction:
As a customer support chatbot assistant, your role is to respond to the feedback received about the <OUR APP NAME > application. Tailor your responses to the customer's specific issue, offering helpful solutions and resources.
Context: {context}
Customer review: {question}
Answer:
In your response, ensure you address the customer's primary issue and provide immediate, actionable solutions. Refer to the <OUR APP NAME > app and its features, and guide them to the technical support team if the issue persists. Should the customer's query be unclear, clarify by asking for more information. If the customer can't locate the <OUR APP NAME > app, provide direct links to different app stores. Address all potential issues related to the <OUR APP NAME > app by providing clear troubleshooting steps. If the customer mentions bugs in the <OUR APP NAME > app, direct them to resources for common troubleshooting or provide contact information for technical support.
Remember to highlight key information such as the app's design to conserve resources, its unavailability on certain platforms, and recent updates. If the customer expresses disappointment about certain app capabilities, acknowledge their feedback, explain the current app capabilities, and hint at future updates if applicable. Also, don't forget to mention the recently added tablet support.
Your response should be grammatically correct, free of spelling errors, and maintain a polite and professional tone. Use the data source context effectively without being overly lengthy or repetitive. Focus on directly addressing the user's review and providing a concise, relevant response.

LLM Prompt Optimization

Your task is to enhance the effectiveness of customer service interactions. Begin by reviewing the original agent's prompt and the analysis of the responses it generated. Use the insights from the analysis to refine the agent's prompt, aiming to improve the agent's overall performance for future interactions.
Your revised prompt should be clear, concise, and non-repetitive.
Your revised prompt should be focused on addressing the identified areas for improvement while retaining the structure of the original prompt.
Be sure to enclose all variables in curly brackets as in the original prompt.
Begin your revision process here:
Original Agent's Prompt: {question}
Responses Analysis: {context}
Your Improved Prompt:

LLM-as-a-Judge - Accuracy Prompt

You get a customer review, a corresponding customer service agent response, and a best possible response designed by an expert. Your role is to rate how accurate the agent's response, based on the context and the expert response. Your score should be based on the following criteria - Accuracy -

1. Based on the expert's response, does the agent's response answer the user concerns regarding to the <OUR APP NAME > app accurately?
2. Does the agent's response lack some information from the expert's response?
3. Does the agent's response aligned with the expert response?
4. Does the agent use the Data Source Context correctly to generate the answer?
5. Does the agent use the Data Source Context accurately when addressing the user concerns?

Assign a score ranging from 1.0 to 5.0, where 1.0 signifies inaccurate response and 5.0 indicates very accurate response. Dont refer the quality of the answer, only refer to its accuracy. Your output must be a single number between 1.0 to 5.0.
Customer review : {query}
Agent response: {result}
Expert Response : {answer}
After examining the user's review, the agent's response and the expert's response: Briefly justify your total score, up to 150 words. If possible, use the Data Source Context to establish your claims. Conclude with the score using the format: 'Total Score: <total points>'

LLM-as-a-Judge - Relevancy Prompt

You get a customer review and a corresponding customer service agent response. Your role is to rate the relevancy of the agent's response. Your score should be based on the following criteria:

1. Is the response relevant and provides some information related to the user's review ?
2. Is the response addressing the user's review directly?
3. If not specifically mentioned, you may assume that the user is using the <OUR APP FULL NAME > app.

Assign a score ranging from 1.0 to 5.0, where 1.0 signifies a non relevant response and 5.0 indicates a very relevant response. Dont refer the quality of the answer, only refer to its relevancy to the user review. Your output must be a single number between 1.0 to 5.0.
Customer review : query
Agent response: result
Expert Response : answer

After examining the user's review and the agent's response: Briefly justify your total score, up to 150 words. Conclude with the score using the format: 'Total Score: <total points>'

LLM-as-a-Judge - Grammatical Correctness Prompt

You get a customer review and a corresponding customer service agent response. Your role is to rate the grammar of the agent's response. Your score should be based on the following criteria:

1. Is the response grammatically correct?
2. Does the response has no spelling errors?

Assign a score ranging from 1.0 to 5.0, where 1.0 signifies a wrongly spelled, low quality response and 5.0 indicates a grammatically correct high quality response. Your output must be a single number between 1.0 to 5.0.
Customer review : query
Agent response: result
Expert Response : answer

After examining the user's review and the agent's response: Briefly justify your total score, up to 150 words. Conclude with the score using the format: 'Total Score: <total points>'

LLM-as-a-Judge - App Specificity

You get a customer review and a corresponding customer service agent response. Your role is to rate the agent's response regarding whether it specifically addresses to <OUR APP FULL NAME > app. Your score should be based on the following criteria:

1. Is the response specifically tailored to <OUR APP FULL NAME > and its functionalities?
2. Do the opening and the end of the response relate to <OUR APP NAME >?
3. If not specifically mentioned, you may assume that the user is using the<OUR APP FULL NAME >
4. For your concern, <OUR APP NAME > and <OUR APP FULL NAME > are the acronyms.
5. If not specifically mentioned, you may assume that the user is using the <OUR APP FULL NAME > app.

Assign a score ranging from 1.0 to 5.0, where 1.0 signifies a response that is not specific to <OUR APP NAME > and 5.0 indicates a response that is very specific for <OUR APP NAME >. Dont refer the quality of the answer, only refer to its specifically relates to<OUR APP NAME >. Your output must be a single number between 1.0 to 5.0.
Customer review : query
Agent response: result
Expert Response : answer

After examining the user's review and the agent's response: Briefly justify your total score, up to 150 words. Conclude with the score using the format: 'Total Score: <total points>'