User Reviews as a Source for Usability Requirements: A Precursor Study on Using Large Language Models

Cedric Wellhausen[1], Laura Reinhardt[1] and Kurt Schneider[1]

[1]Leibniz University Hannover, Software Engineering Group, Hannover, Germany

Abstract

[Context and motivation] It is known that user-centered approaches to requirements engineering in general lead to a better suited product for the end-users. LLM4RE provides promising approaches to support the requirements elicitation process (e.g. classification of requirements). **[Question/problem]** Previous approaches focus on Machine-Learning (ML) or Deep-Learning (DL) aspects, which require intensive training with a large amount of manually labeled data. LLMs, on the other hand, are pre-trained on large amounts of user-generated text data, enabling a user-centric workflow to analyze requirements. **[Principal ideas/results]** In this paper, we explore the possibility of exploiting the improved natural language understanding of LLMs, rather than strict ML classification, together with the mass extraction of user reviews to analyze if the performance of LLMs in understanding user reviews is comparable to the performance of human raters. This enables a quick and cheap workflow for development teams to gather and process their user's requirements. **[Contribution]** This paper provides three major contributions: (1) We provide a completely coded dataset of 300 user reviews containing usability-relevant aspects from three different types of apps, that were labeled by two human raters and by an LLM. (2) We build an initial prompt, based on two prompt engineering iterations and specifically developed coding guidelines derived from the 10 Nielsen Usability Heuristics, for LLMs to filter usability relevant user reviews. (3) We determine that LLMs are generally able to recognize usability as a non-functional requirement in user reviews, in terms of their F-score, but the performance and reliability is strongly dependent on the prompt.

Keywords

Usability, LLM, Requirements Engineering

1. Introduction

The increasing functional complexity of modern software products presents major usability challenges. Although integrating user-centred approaches into requirements engineering can mitigate these issues and lead to a better suited product for end-users [1], directly involving users remains a resource-intensive process that is often unfeasible for development teams. To bridge the gap between development teams and users, Crowd-based Requirements Engineering (CrowdRE) leverages publicly available feedback from forums and app stores to indirectly capture user needs from user feedback [2].

Machine-Learning and Deep-Learning approaches for classifying requirements are frequently combined with CrowdRE or crowd-sourcing methods [3, 4, 5]. One significant limitation of these approaches in classifying user needs out of user feedback is, that these approaches often rely on large, manually labeled datasets, which creates a significant bottleneck for rapid development cycles [3, 4, 5]. Nowadays, new approaches involving Natural Language Processing (NLP4RE) and Large Language Models (LLM4RE) gain more attention, due to the fact, that LLMs are able to process a large amount of information, without requiring any additional training data for most purposes [6]. Despite the rise of Natural Language Processing (NLP4RE) [3] and Large Language Models (LLM4RE) [4] in requirements engineering tasks, a critical research gap remains: combining the effectiveness of modern LLM language understanding with the user-centric methods of CrowdRE.

In this work, we address the potential of LLMs to extract and process user requirements from mass review data without the need for extensive task-specific training. LLMs can be instructed with natural language, where the instructor gives an LLM a so-called *prompt* [7]. We want to compare if

Joint Proceedings of REFSQ-2026 Workshops, Doctoral Symposium, Posters & Tools Track, and Education and Training Track. Co-located with REFSQ 2026. Poznan, Poland, March 23-26, 2026.

cedric.wellhausen@gmail.com (C. Wellhausen); laura.reinhardt@inf.uni-hannover.de (L. Reinhardt); kurt.schneider@inf.uni-hannover.de (K. Schneider)

the performance of an specifically prompted LLM is comparable to the performance of human coders to identify usability-relevant user reviews. Furthermore, the LLM will be analyzed in context of their reliability to identify usability-related user reviews with an tailored prompt, specifically developed for these kind of tasks. This work will test the mass elicitation in light of usability requirements which can (i) be very nuanced and (ii) play an important role in product marketing and user adoption.

2. Background and Related Work

2.1. Usability

Usability by ISO 9241-11 is defined as the extent to which a product, including mobile apps, can be used by specified users to achieve their specified goals [8]. This also includes factors such as learnability and satisfactory in use [8]. Furthermore, usability can be defined as the general capability of an entity to being used [9, 10]. Due to the widespread definitions and applicability of usability in all kinds of systems, it is quite difficult to establish a single, universally accepted method, necessitating an iterative and context-dependent approach to evaluation. One of the more established methods to evaluate the usability of software systems is the *10 Usability Heuristics* by Nielsen [11, 12]. These heuristics include a set of ten general principles on how to design user interfaces [11, 12]. Additionally, usability problems in real systems can be derived from these heuristics [13].

2.2. Prompt Engineering

Prompt Engineering is an emerging field that aims to produce an input for a Large Language Model (LLM) such that the LLM will produce a desired output when given the input [14]. Such an input is then called a *prompt*. A textual prompt is a set of instructions, given to a LLM, to customize, enhance and refine its capabilities [14, 15]. This adaptability of prompts as an input for LLMs is inherently different from traditional machine learning, where model retraining is often required for specific tasks [16]. Generating a fitting prompt for a desired output can not be perfectly formalized but is approached with a set of techniques that have been shown to work well in practice (e.g. iterative prompt engineering [17], chain-of-thought prompting [18]). One technique, iterative prompting, involves direct human feedback, also known as human-in-the-loop, to create an initial draft of a prompt that is then further refined by the humans [17]. The chain-of-thought prompting technique aims to improve LLMs' outputs by prompting an LLM to describe its intermediate steps while performing tasks, thereby making its approach transparent for further analysis [17, 18].

2.3. Related Work

Bakiu et al. [1] have developed a Machine-Learning-based method for extracting usability aspects from user reviews. To this end, a classifier was trained with manually classified reviews from the categories *Software* and *Video Games*. To do this, they used four sets of usability dimensions, including the five dimensions of Nielsen [19] as one of the sets. The findings revealed, that only analyzing user reviews can not be replaced by already existing methods. Furthermore, their method to extract usability aspects was provisionally evaluated with the result that further and larger studies are necessary to gain an accurate picture of its effectiveness.

Hedegaard et al. [20] conducted a study collecting reviews from the categories *Software* and *Video Games*. They used 2972 reviews from the video game sector and 520 reviews from the software sector and used different models to extract usability aspects. Therefore they used one model (CLASSICUA) based on the usability definition according to Nielsen [19] and another model (FREQUENT) based on terms that are frequently associated with usability. The CLASSICUA classification resulted in usability-relevant aspects being mentioned in more than 40% of the reviews. FREQUENT achieved a value of up to 30%.

Previous research has concentrated exclusively on machine-learning (ML) approaches to the classification of usability-relevant aspects. However, such approaches require a substantial quantity of labeled training data. In the context of our research, there is a clear objective to transition the perspective from the utilization of resource-intensive machine learning algorithms to the deployment of more accessible Large Language Models (LLMs). While ML is only able to simply classify usability aspects on simple keywords (e.g. ”slow” or ”error”), LLMs are able to understand contextual information and informal language like emojis (e.g. ”I had to click the Back-Button three times :-/”). This ability facilitates the identification of usability aspects in a more diverse range of written statements. In addition, LLMs have the capacity to justify their decisions, a capability that is absent in the case of ML, which provide solutions without offering any reasoning to support their decisions. Our work is also supposed to analyze different types of software, that are more integrated in daily life. Therefore, we not only included reviews from software in general, but rather specific software used in maps and navigation, music-composition or E-commerce [21].

3. Research Design

3.1. Research Questions

Our research in the work is framed by the following research questions:

RQ1: Is the performance of an LLM, when prompted with a specifically tailored prompt to identify usability-related user reviews, comparable to the performance of a human coder?
RQ2: What is the reliability of an LLM for identifying usability-related user reviews in a random dataset of user generated app reviews in case of specifically tailored prompts?

In this paper, we aim to investigate whether the LLM gpt-4.1 by OpenAI can be used as an effective and resource efficient tool to support requirements elicitation in the context of CrowdRE. Our focus lies on the non-functional requirement usability as this allows us to (1) test the LLM in a controlled environment that can be defined in terms of norms and heuristics and (2) to see in what extent the LLM understands user reviews, as these are in general unstructured and extracting usability-related information has proven difficult [1].

3.2. Data Collection

To test the capabilities of the LLM on real-world data, we collected a dataset of 300 user reviews. The 300 user reviews originate from the Google Play-Store and are evenly distributed over three apps: *BlitzerDE*, *Lidl Plus*, and *FL Studio*. The three apps were selected based on the criteria of user numbers, ratings and number of ratings. Both criteria were intended to maximize the diversity of individual reviews such that a broad range of opinions could be covered. Furthermore, apps from different software types and specific categories were selected in order to obtain a broad picture across various domains. Table 1 shows the ratings, user counts and specific software type for all three apps.

Table 1
Overview of the apps which served as sources for user reviews, their respective Google Play-Store analytical data and their type of software.

App	Rating	User Count	Ratings Count	Software Type [21]
BlitzerDE	4.7	10 million	>120k	Information display and transaction entry (maps and navigation)
Lidl Plus	3.8	100 million	>1.70 million	Information display and transaction entry (E-commerce)
FL Studio	4.0	1 million	>40k	Computation-dominant (music composition)

Besides the aforementioned criteria, we also included the genre and intended target audience in our decision. All three apps are targeted towards adult users but in very different scenarios. While BlitzerDE is used in an automotive-centric scenario, Lidl Plus's users will use the app before and while shopping for e.g. groceries. FL Studio the other hand is an app that is aimed at musicians to help them produce music. As these different scenarios affect how the usability of the app is perceived by the users, we assume the user reviews will contain vastly different opinions and feedback. The goal is to construct a dataset that has a high diversity of user reviews which then can be used to test the LLM with as little bias towards certain kinds of user reviews as possible.

3.3. Data Analysis

The data analysis process of this paper is divided into two phases. The first phase serves the purpose of (1) building a fully labeled dataset which can be used to evaluate the performance of an LLM, and (2) establishing a performance baseline that represents the performance of usability-experts and can be used to evaluate the LLM. The second phase uses the results of the first phase to then answer both research questions RQ1 and RQ2.

First Phase: Figure 2 shows the process that was developed for the first phase. Labeling the dataset began with establishing a set of indicators for usability. For this, we used the *10 Usability Heuristics* according to Nielsen [11, 12] as well as a few indicators we defined as not usability-relevant. These are: (i) Feature-Requests, (ii) overlap with adjacent non-functional requirements such as accessibility and compatibility, and (iii) vaguely formulated statements for which too much interpretation by the raters would haven been necessary. We decided to use the 10 Usability Heuristics as they define usability very detailed when compared to other definitions like the ISO 9241 [8].

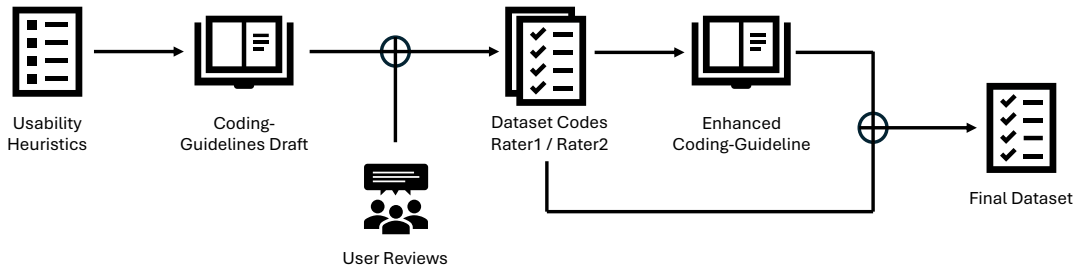

Figure 1: Data collection process for building a fully labeled dataset.

The labeling of the user reviews was performed by two raters, both with multiple coding experience in context of usability and explainability, which is why we believe that the results were not influenced by the quality of the raters. Individual user review were rated in the binary format of *true* and *false*. We decided to code a user review with *true*, if at least one usability-relevant statement is included in the review, otherwise we rate it as *false*. For example, the review "I find the app rather cumbersome; some things could have been made easier" would be rated as *true* because it relates to the heuristic "Flexibility and Efficiency of Use." On the other hand, the review "Google Calendar works perfectly with aCalendar" would be rated *false* because it does not directly address usability, but rather compatibility.

This first draft of coding guidelines was used by two raters in a pre-coding round. The pre-coding only looked at the first 20 user reviews of the *BlitzerDE* app. We noted some differences in the understanding of the guidelines between raters. These issues were resolved by adding examples to the heuristics to the draft guidelines. After the draft guidelines were established, both raters fully labeled the dataset of 300 user reviews. To finalize the coding guidelines, the raters compared their results to each other. The comparison showed further issues with the draft guidelines. These are e.g. (a) bug-related reviews, (b) the handling of usability-problems which stem from outside of the apps context, e.g. Android, and (c) misleading user reviews. The coding guidelines were updated according to these issues and further examples were provided. With these enhanced coding guidelines, we

conducted a second round of coding and calculated the interrater-reliability. To fulfill purpose (1), the last differences in the coding of both raters have been finalized into a single rating. In the finalization of our coding process, we discussed each user review with differing codes. We fulfill purpose (2) of the data collection process by comparing the results of the LLM with our final dataset and apply the following metrics: the interrater-reliability which treats the LLM as a human rater and allows direct comparisons to human performance, and the F-Score which is a common metric for binary classification performance and gives insights over the absolute ability of the LLM and the chosen prompt to label usability-relevant user reviews.

Second Phase: The second phase of our data analysis answers research questions RQ1 and RQ2. For this, the LLM is used to label the full dataset of 300 user reviews. To answer RQ1, the LLM is evaluated according to the ground-truth established in the data collection in terms of (M1) precision, (M2) accuracy, and (M3) interrater-reliability. Metrics (M1) and (M2) allow us to investigate if the LLM generally is able to distinguish usability-related reviews in a random sample, whereas metric (M3) gives insights about the LLMs reliability for identifying usability-related reviews when compared to human usability-experts.

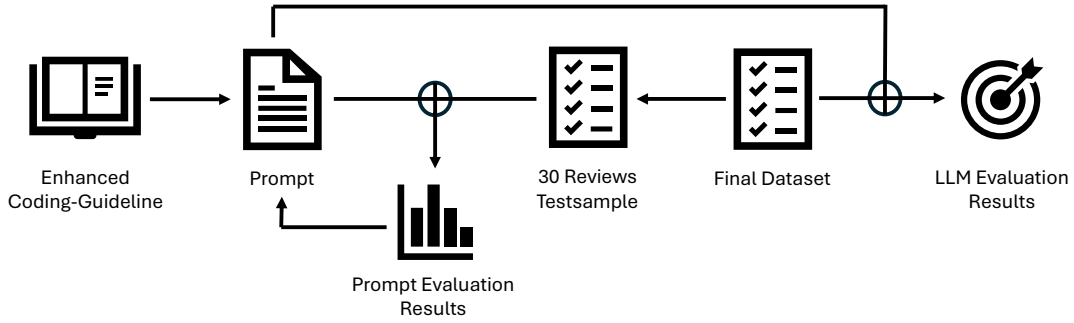

Figure 2: Data analysis process with the final dataset and an initial prompt as input to the iterativ prompt evaluation process, which results in a new and optimized prompt that can be fed back into the process to further refine it.

The process consists of two parts (i) generating and evaluating a prompt based on the enhanced coding guidelines from the data collection, and (ii) evaluating the LLM based on an specifically tailored prompt when tasked to label the final dataset from the data collection process. The specifically tailored prompt was generated through an iterative process to achieve a good performance. Based on the enhanced coding guidelines, an initial prompt was designed. To create the initial prompt, we used chain-of-thought prompting to give the LLM a clear list of instructions on how to work with the data. However, this first version solely served the purpose of being a template for further iterations. To evaluate prompt-iterations, the LLM was given the current iteration of the prompt and a subset of the fully labeled dataset. This test sample consists of 30 (10%) randomly selected user reviews with 10 reviews from each app. For this paper, three iterations were conducted with the last iteration resulting in the final prompt. While evolving the prompt, we decided against using few-shot prompting. Instead, we added notes to the prompt that more clearly specified our goals, while being general enough to not cause overfitting inside the prompt.

After three iterations of adjusting the specifications inside the prompt, the final prompt and the full dataset were then used to evaluate the LLM according to the defined metrics (M1), (M2), and (M3).

3.3.1. Data Availability Statement

To ensure the transparency and verifiability of our research, we provide the following data in our supplementary material [22]: Firstly, we provide the Python script used to collect the user reviews from the Google Play-Store. We also provide the coding guidelines, as well as the full dataset containing all

the tables with user reviews and their ratings in terms of usability-relevant aspects. Lastly, we provide all prompts that were used to detect usability aspects in user reviews with an LLM.

4. Results

4.1. First Phase: Manually Labeled User Reviews

Since the coding process involved two coders and the data was labeled on a nominal scale, we report the interrater reliability using Cohen's Kappa κ [23]. According to Landis and Koch [24] we achieved a substantial agreement for *BlitzerDE* (Cohen's kappa κ = 0.76). *Lidl Plus* also achieved a substantial agreement (Cohen's kappa κ = 0.70) and *FL Studio* also achieved a substantial agreement (Cohen's kappa κ = 0.61) in our final coding round.

Table 2
Interrater Agreement (Cohen's Kappa κ) for the manual detection.

App	Cohen's Kappa κ	Agreement according to Landis et al. [24]
BlitzerDE	0.76	Substantial
Lidl Plus	0.70	Substantial
FL Studio	0.61	Substantial

After resolving all conflicts, from the 300 user reviews, the two raters were able to extract 148 (49.3%) user reviews in total which involved at least one usability aspect using the predefined coding guidelines. The final dataset contained 56 usability-relevant user reviews from *BlitzerDE*, 54 user reviews from *Lidl Plus*, and 38 reviews from *FL Studio*.

4.2. Second Phase: Automatic Labeling of User Reviews

We chose to use gpt-4.1 by OpenAI to evaluate the user reviews as this was the most recent model at the time of conducting the evaluation and had a well documented API. This section presents (1) the results of the prompt generation which was tested on a randomized sample of 30 user reviews from the full dataset, and (2) the final LLM evaluation results of a full-scale test on the dataset.

Table 3
Results of the three prompt iterations on the Testsample.

Iteration	BlitzerDE (n=10)			Lidl Plus (n=10)			FL Studio (n=10)		
	Precision	Recall	F_1-Score	Precision	Recall	F_1-Score	Precision	Recall	F_1-Score
Iteration 1	0.33	1.0	0.5	0.67	1.0	0.8	0.3	1.0	0.46
Iteration 2	0.375	1.0	0.55	0.75	1.0	0.86	0.33	1.0	0.5
Iteration 3	0.67	0.67	0.67	0.83	0.83	0.83	0.75	1.0	0.86

The initial prompt has been revised two times and has been evaluted three times at all stages. For our method of improving the initial prompt iteratively, we report the values of the Precision, Recall, and F_1-Score in Table 3. The values for each of the three apps are calculated separately.
Our final evaluation on the full dataset with the gpt-4.1 resulted in the LLM identifying 173 (58%) user reviews as usability-relevant, with 61 user reviews from *BlitzerDE*, 68 reviews from *Lidl Plus* and 44 user reviews from *FL Studio*. Table 4 shows the Precision, Recall, and F_1-Score for this evaluation. In total, the LLM correctly labeled 239 (79.6%) user reviews.
For the interrater reliability with the LLM, we used the final coding result table of the two raters with all resolved conflicts and the coding result table generated by the LLM using the final prompt. Since this coding process also involved two coders, the combined codes by the human raters from the third coding round and the codes determined by the LLM, and the data also was labeled on a nominal scale,

Table 4
Results of the final evaluation on the full dataset.

BlitzerDE (n=100)			Lidl Plus (n=100)			FL Studio (n=100)		
Precision	Recall	F_1-Score	Precision	Recall	F_1-Score	Precision	Recall	F_1-Score
0.79	0.86	0.82	0.74	0.93	0.82	0.73	0.84	0.78

we report the interrater agreement using Cohen's Kappa κ [23]. According to Landis and Koch [24] the human codings combined with the LLM codings achieved a fair agreement for *BlitzerDE* (Cohen's kappa κ = 0.34). *Lidl Plus* achieved a moderate agreement (Cohen's kappa κ = 0.55) and *FL Studio* achieved a substantial agreement (Cohen's kappa κ = 0.63).

Table 5
Interrater Agreement (Cohen's Kappa κ) for the detection with an LLM and the final coding results of the third iteration. The κ is calculated between the labels in the final dataset and the labels of the LLM.

App	Cohen's Kappa κ	Agreement according to Landis et al. [24]
BlitzerDE	0.34	Fair
Lidl Plus	0.55	Moderate
FL Studio	0.63	Substantial

Further, we report the observed values for when the two raters were certain in their final rating compared to the LLM labeling correctly according to the final rating. Figure 3 shows the distribution of the four possible combinations for the accumulated values of the three apps. A rating is considered certain if both raters agreed on the final rating. The LLM labeling is considered correct if it equals the final rating in the fully labeled dataset. The χ^2-Test results on these values in a p-value of 0.35. The observed values are therefore not statistically significant at $p < 0.05$.

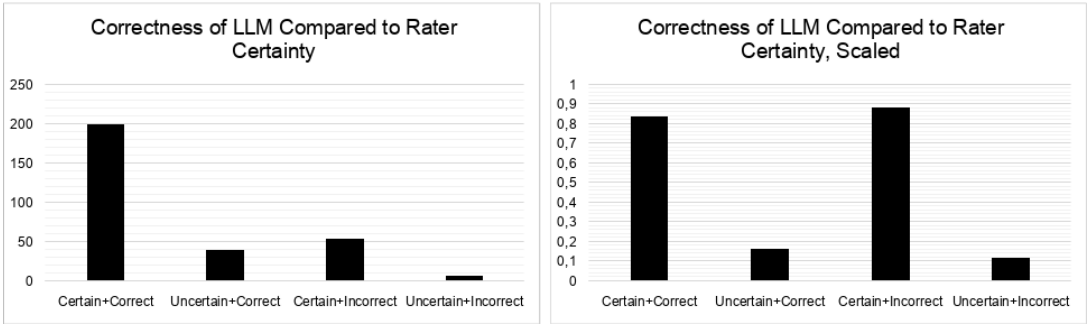

Figure 3: Distribution of the cross-examination of certainty of the raters vs. the correctness of the LLM's evaluation. Left side: Absolute values. Right side: Relatively values, scaled by 239 (cases of LLM being correct) and 61 (cases of LLM being incorrect).

5. Discussion

5.1. Answering the Research Questions

RQ1: Is the performance of an LLM, when prompted with a specifically tailored prompt to identify usability-related user reviews, comparable to the performance of a human coder? As can be seen in Table 4, the LLM has achieved high values for the Recall measurements, which indicates a high correspondence with our findings of usability-relevant user reviews. Although not perfect, this shows that a significant portion of usability-relevant user reviews can be identified by an LLM with a prompt that is specifically tailored over multiple iterations. When combined with the

Precision score, which indicates that the LLM's identified usability relevant user reviews were correct assessments, the LLM shows to be more permissive with its labeling than the human raters.

RQ2: What is the reliability of an LLM for identifying usability-related user reviews in a random dataset of user generated app reviews in case of specifically tailored prompts? Table 4.2 reports a mixed reliability when compared to the human raters. For BlitzerDE and Lidl Plus, the human raters had a significantly better agreement than the LLM with the fully labeled dataset. However, the LLM performed slightly better for the FL Studio partition of the dataset. With the LLM achieving Substantial and Moderate agreement for FL Studio and Lidl Plus respectively, a prompt with just three iterations already achieves reliable results. However, taking Figure 3 into perspective, the LLM's correctness in labeling the user reviews does not correlate with the raters being certain in their rating. The LLM therefore doesn't reliably reproduce the mental model of the human raters and produces errors in different and unexpected places. This issue may be alleviated by improving the prompt in such a way, that the thought process of an usability-expert can be recreated by the LLM.

5.2. Threats to Validity

We discuss the threats to the validity of this work in accordance with Wohlin et al. [25]:

Construct Validity. Online user reviews are generally carried by emotions and are therefore easily influenced by both positive and negative emotions. To minimize this influence on the fully labeled dataset, we excluded to broadly formulated statements from being labeled as usability. We only focused on statements that directly contained a reference to one of the heuristics. Negative experiences with software are also more likely to be reported than good or neutral experiences. As negative experiences are commonly linked to bugs or unexpected behavior, we excluded bug-related statements from being labeled as usability-relevant, even though these could be classified under the heuristic *Error Prevention*.

Internal Validity. Users may report problems as a review to an app, that actually stems from other components of the system, like the operating system or hardware issues. To mitigate this, we only included statements that are linked to the apps usability. Statements that were clearly related to other components of the system and only affected the apps usability as a side effect, were excluded from being labeled usability-relevant. Updates may introduce new bugs or features to an app. This can lead to a change in usability or a wave of both positive feedback, praising the new feature as well as negative feedback, criticizing the new feature or reporting bugs. To mitigate this selection bias, we chose to use the sorting by *most relevance* for collecting user reviews, as the other options *newest* and *ratings* would introduce a selection bias for either the aforementioned problem as well as only retrieving the best or worst feedback. Another important factor is that the prompt to evaluate the user reviews was given to the LLM in english, but the user reviews are in german. As we did no direct comparison with a german prompt, mixing both english and german language for the input to the LLM, side-effects are generally possible. However, LLMs and specifically GPT-4.1 achieve high benchmarking scores in specifically multilingual benchmarking tests, such as the *Multilingual MMLU* [26].

Conclusion Validity. The conclusions drawn from the labeled dataset are heavily influenced by the quality of the raters. As there were only two raters, a significant shift in the results could be seen if one rater had significantly more or less familiarity with usability. We believe that the results were not influenced by the quality of the raters, as we designed the guidelines in a collaborative way that included detailed discussions of both the usability heuristics and when to apply them, as well as possible edge-case scenarios. This is supported by our substantial agreement in the final round of coding.

External Validity. Our results are not directly generalizable to the entirety of apps and LLMs. This is due to both the limited amount of apps we analyzed as well as the limited amount of LLMs we

experimented on. As can be seen in our results, even in our small sample of apps, LLM performance may vary drastically. It is unclear to us if the observed effects are amplified when looking at a larger frame or eventually even out. For this work, we took apps with vastly different usage scenarios to gather a diverse set of user reviews to increase the generalization as much as possible.

6. Conclusion and Future Work

In this paper, we examined if using LLMs together with CrowdRE approaches can be an effective method for user-centric requirements engineering. We conclude that LLMs, although not perfect, should be considered as a viable option in requirements engineering for labeling unstructured data like user reviews. Our data shows that the reliability between the human raters and the LLM is too low as for the LLM to replace human raters yet. We still argue that the resource overhead (i.e. training) of using more traditional Deep-Learning or Machine-Learning approaches isn't practical for most requirements engineers, when an LLM can achieve acceptable results with significantly less overhead. We found that generating a tailored prompt is one of the defining aspects for a successful labeling task. In our results, the LLM tends to be slightly more permissive, which we do not interpret as a problem. In practice, we consider gathering a limited amount of irrelevant user review as better than missing a relevant user review.

This work serves as a precursor to further research into the LLM integration into the CrowdRE framework. Our results provide a fully labeled dataset which can be used as a performance baseline. In future work, we aim to increase the LLMs reliability with more sophisticated prompt engineering methodologies, such as few-shot prompting. We also want to compare the traditional methods of Deep-Learning and Machine-Learning with LLMs and optimized prompts. This in turn will give more detailed answers on the question of when to use which technology. It is also possible to take a more detailed look at different large-language models. As LLMs are gaining more attention, different companies train new models with different capabilities, that may be leveraged in CrowdRE.

Declaration on Generative AI

During the preparation of this work, the authors used OpenAI's GPT-4.1 in order to label user reviews, which were then further analyzed. The authors did not use text produced by generative AI for this work.

References

[1] E. Bakiu, E. Guzman, Which feature is unusable? detecting usability and user experience issues from user reviews, in: 2017 IEEE 25th International Requirements Engineering Conference Workshops (REW), 2017, pp. 182–187. doi:10.1109/REW.2017.76.

[2] E. Groen, Crowd-Based Requirements Engineering, Doctoral thesis 2 (research not uu / graduation uu), Universiteit Utrecht, 2025. doi:10.33540/3091.

[3] L. Zhao, W. Alhoshan, A. Ferrari, K. J. Letsholo, M. A. Ajagbe, E.-V. Chioasca, R. T. Batista-Navarro, Natural language processing for requirements engineering: A systematic mapping study, ACM Comput. Surv. 54 (2021). URL: https://doi.org/10.1145/3444689. doi:10.1145/3444689.

[4] M. A. Zadenoori, J. Dąbrowski, W. Alhoshan, L. Zhao, A. Ferrari, Large language models (llms) for requirements engineering (re): A systematic literature review, 2025. URL: https://arxiv.org/abs/2509.11446. arXiv:2509.11446.

[5] M. Unterbusch, M. Sadeghi, J. Fischbach, M. Obaidi, A. Vogelsang, Explanation needs in app reviews: Taxonomy and automated detection, 2023, pp. 102–111. doi:10.1109/REW57809.2023.00024.

[6] F. Wei, R. Keeling, N. Huber-Fliflet, J. Zhang, A. Dabrowski, J. Yang, Q. Mao, H. Qin, Empirical study of llm fine-tuning for text classification in legal document review, in: 2023 IEEE International

Conference on Big Data (BigData), 2023, pp. 2786–2792. doi:10.1109/BigData59044.2023.10386911.

[7] J. Dąbrowski, W. Cai, A. Bennaceur, B. Nuseibeh, F. Alrimawi, Intelligent agents for requirements engineering: Use, feasibility and evaluation, in: 2025 IEEE 33rd International Requirements Engineering Conference (RE), 2025, pp. 535–543. doi:10.1109/RE63999.2025.00064.

[8] ISO, ISO 9241-110:2020 Ergonomics of human-system interaction — Part 110: Interaction principles, International Standards Organisation (2024).

[9] N. Bevan, J. Carter, S. Harker, Iso 9241-11 revised: What have we learnt about usability since 1998?, in: M. Kurosu (Ed.), Human-Computer Interaction: Design and Evaluation, Springer International Publishing, Cham, 2015, pp. 143–151.

[10] D. Quiñones, C. Rusu, How to develop usability heuristics: A systematic literature review, Computer Standards & Interfaces 53 (2017) 89–122. URL: https://www.sciencedirect.com/science/article/pii/S0920548917301058. doi:https://doi.org/10.1016/j.csi.2017.03.009.

[11] J. Nielsen, 10 usability heuristics for user interface design, 1994. URL: https://www.nngroup.com/articles/ten-usability-heuristics/, last accessed: 01/12/2026.

[12] J. Nielsen, Enhancing the explanatory power of usability heuristics, in: Proceedings of the SIGCHI conference on Human Factors in Computing Systems, 1994, pp. 152–158.

[13] J. Nielsen, Enhancing the explanatory power of usability heuristics, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '94, Association for Computing Machinery, New York, NY, USA, 1994, p. 152–158. URL: https://doi.org/10.1145/191666.191729. doi:10.1145/191666.191729.

[14] J. White, Q. Fu, S. Hays, M. Sandborn, C. Olea, H. Gilbert, A. Elnashar, J. Spencer-Smith, D. C. Schmidt, A prompt pattern catalog to enhance prompt engineering with chatgpt, 2023. URL: https://arxiv.org/abs/2302.11382. arXiv:2302.11382.

[15] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, G. Neubig, Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing 55 (2023). URL: https://doi.org/10.1145/3560815. doi:10.1145/3560815.

[16] P. Sahoo, A. K. Singh, S. Saha, V. Jain, S. Mondal, A. Chadha, A systematic survey of prompt engineering in large language models: Techniques and applications, 2025. URL: https://arxiv.org/abs/2402.07927. arXiv:2402.07927.

[17] The prompt report: A systematic survey of prompt engineering techniques, 2025. URL: https://arxiv.org/abs/2406.06608. arXiv:2406.06608.

[18] J. Wei, X. Wang, D. Schuurmans, M. Bosma, b. ichter, F. Xia, E. Chi, Q. V. Le, D. Zhou, Chain-of-thought prompting elicits reasoning in large language models, in: S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, A. Oh (Eds.), Advances in Neural Information Processing Systems, volume 35, Curran Associates, Inc., 2022, pp. 24824–24837. URL: https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf.

[19] J. Nielsen, Usability Engineering, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1994.

[20] S. Hedegaard, J. G. Simonsen, Extracting usability and user experience information from online user reviews, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '13, Association for Computing Machinery, New York, NY, USA, 2013, p. 2089–2098. URL: https://doi.org/10.1145/2470654.2481286. doi:10.1145/2470654.2481286.

[21] A. Forward, T. C. Lethbridge, A taxonomy of software types to facilitate search and evidence-based software engineering, in: Proceedings of the 2008 Conference of the Center for Advanced Studies on Collaborative Research: Meeting of Minds, CASCON '08, Association for Computing Machinery, New York, NY, USA, 2008. URL: https://doi.org/10.1145/1463788.1463807. doi:10.1145/1463788.1463807.

[22] C. Wellhausen, Supplementary material to user reviews as a source for usability requirements, 2026. URL: https://figshare.com/collections/Supplementary_Material_to_User_Reviews_as_a_Source_for_Usability_Requirements/8256262/2. doi:10.6084/m9.figshare.c.8256262.v2.

[23] J. Cohen, A coefficient of agreement for nominal scales, Educational and Psychological Measurement 20 (1960) 37–46. doi:10.1177/001316446002000104.

[24] J. R. Landis, G. G. Koch, The measurement of observer agreement for categorical data, Biometrics 33 (1977) 159–174. doi:https://doi.org/10.2307/2529310.
[25] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, A. Wesslén, et al., Experimentation in software engineering, volume 236, Springer, 2012.
[26] OpenAI, Introducing gpt-4.1 in the api, 2025. URL: https://openai.com/index/gpt-4-1/, last accessed: 02/17/2026.