Identifying and Classifying User Requirements in Online Feedback via Crowdsourcing

Martijn van Vliet[1], Eduard C. Groen[1,2] ✉, Fabiano Dalpiaz[1], and Sjaak Brinkkemper[1]

[1] Dept. of Information and Computing Sciences, Utrecht University, Netherlands
{m.vanvliet, f.dalpiaz, s.brinkkemper}@uu.nl
[2] Fraunhofer IESE, Kaiserslautern, Germany
eduard.groen@iese.fraunhofer.de

Abstract. [Context and motivation] App stores and social media channels such as Twitter enable users to share feedback regarding software. Due to its high volume, it is hard to effectively and systematically process such feedback to obtain a good understanding of users' opinions about a software product. **[Question/problem]** Tools based on natural language processing and machine learning have been proposed as an inexpensive mechanism for classifying user feedback. Unfortunately, the accuracy of these tools is imperfect, which jeopardizes the reliability of the analysis results. We investigate whether assigning *micro-tasks* to crowd workers could be an alternative technique for identifying and classifying requirements in user feedback. **[Principal ideas/results]** We present a crowdsourcing method for filtering out irrelevant app store reviews and for identifying features and qualities. A validation study has shown positive results in terms of feasibility, accuracy, and cost. **[Contribution]** We provide evidence that crowd workers can be an inexpensive yet accurate resource for classifying user reviews. Our findings contribute to the debate on the roles of and synergies between humans and AI techniques.

Keywords: Crowd-based requirements engineering · Crowdsourcing · Online user reviews · Quality requirements · User feedback analysis.

1 Introduction

As a growing body of requirements engineering (RE) literature shows, substantial amounts of online user feedback provide information on user perceptions, encountered problems, suggestions, and demands [3,21,22]. Researchers have predominantly focused on analyzing user feedback about mobile apps. Of the various online sources of user feedback, they have emphasized app stores and Twitter because these readily offer large amounts of user feedback [23].

The amount of feedback typically obtained for an app is too large to be processed manually [12,17], and established requirements elicitation techniques, such as interviews and focus groups, are not suitable for engaging and involving the large number of users providing feedback. Hence, user feedback analysis has

become an additional elicitation technique [13]. Because most user feedback is text-based, natural language processing (NLP) techniques have been proposed to automatically—and thus efficiently—process user feedback [4,22,24,34].

However, although NLP approaches perform well for simple tasks such as distinguishing informative from uninformative reviews, they often fail to make finer distinctions such as feature versus bug, or privacy versus security requirements [5,34]. Also, most NLP techniques focus on functional aspects, while online user feedback has been found to contain much information on software product quality by which users are affected directly [11], such as usability, performance, efficiency, and security. Their correct identification is made more difficult by language ambiguity due to poor writing [34]. Extensive training and expert supervision are required to improve the outcomes of NLP techniques.

We surmise that a *crowdsourcing-based approach* to identifying and classifying user feedback could overcome the limitations of existing approaches that are NLP-based or reliant on expert analysts. The premise is to train crowd workers to perform the classification. Spreading the tagging workload over the members of an inexpensive crowd might make this approach a feasible alternative for organizations, with more accurate results than those obtained through automated techniques. Moreover, since the extraction is done by human actors, the results may in turn be used as training sets for NLP approaches [12,17,30].

The challenge is that the quality of the annotation results largely depends on the knowledge and skills of the human taggers. A crowdsourcing setting offers access to many crowd workers, but they are not experienced in requirements identification or classification. Hence, we employ strategies from the crowdsourcing field [18], including the provision of *quick training* to the workers [7], simplification of their work in the form of *micro-tasks*, and the use of redundant annotators to filter out noise and to rely on the predominant opinion.

Our main research question is: "*How can a method that facilitates the identification of user requirements*[1] *through a sizeable crowd of non-expert workers be constructed?*" Such a method should ease the removal of spam and other useless reviews, and allow laypeople to classify requirements aspects in user reviews. It also needs to be feasible and cost-effective: The quality of the tagging should be regarded sufficiently high by the app development company to justify the investment, also thanks to the time saved by crowdsourcing tasks that would otherwise be performed by employees. We make the following contributions:

1. We present *Kyōryoku*: a crowdsourcing method for eliciting and classifying user requirements extracted from user feedback. Our method aims to allow laypeople to deliver effective outputs by simplifying tasks.
2. We report on a validation of the method performed on a sample of 1,000 app store reviews over eight apps, which attracted a large crowd and provided good results in terms of processing speed, precision, and recall.
3. We provide the results from the crowd workers and our gold standard as an open artifact [33] that other researchers can use for training automated

[1] In this paper, *user requirements* are understood as "a need perceived by a stakeholder", as per one sub-definition of *requirement* in the IREB Glossary [9].

classifiers that rely on machine learning (ML) or for assessing the quality of human- or machine-based classification methods.

Organization. After reviewing related work in Sec. 2, we describe our method in Sec. 3. We present the design of our experiment in Sec. 4, and report and analyze the results in Sec. 5. We review the key threats to validity in Sec. 6, while Sec. 7 presents conclusions and future directions.

2 Related Work

Crowd involvement in RE has been studied by various researchers over the past decade, especially through the proposal of platforms that allow the crowd of stakeholders, users, and developers to actively participate in the communication of needs for creating and evolving software systems [20,29]. The *Organizer & Promoter of Collaborative Ideas* (OPCI; [1]) is a forum-based solution that supports stakeholders in collaboratively writing, prioritizing, and voting for requirements. Through text analysis, initial ideas of the stakeholders are clustered into forums, and a recommender system suggests further potentially relevant forums. Lim and Finkelstein's *StakeRare* method includes an online platform for identifying stakeholders via peer recommendation, and for eliciting and prioritizing the requirements they suggest [20]. *REfine* [29] is a gamified platform based on idea generation and up- / downvoting mechanisms through which stakeholders can express their needs and rank their priority. A similar idea forms the basis of the *Requirements Bazaar* tool [26]. All these platforms offer a public space for stakeholders to interact and express their ideas.

Other researchers have investigated the adequacy of crowd workers in acting as taggers in requirements-related tasks. This has been explored, for example, in the context of user feedback collected from app stores. The Crowd-Annotated Feedback Technique (CRAFT) [16] is a stepwise process that creates micro-tasks for human taggers to classify user feedback at multiple levels: (i) category, e.g., bug reporting vs. feature request; (ii) classification, e.g., whether a bug regards the user interface, error handling, or the control flow; and (iii) quality of the annotated feedback and confidence level of the tagger. CRAFT inspires our work because it aims to provide empirical evidence regarding the actual effectiveness of such annotation techniques in practice. Stanik, Haering and Maalej [30] recently employed crowdsourcing to annotate 10,000 English and 15,000 Italian tweets to the support accounts of telecommunication companies, which in turn served as part of their training set for ML and deep learning approaches.

User feedback classification has seen a rapid rise of automated techniques based on NLP and ML. Research on the automatic classification of feedback has given rise to alternative taxonomies; for example, Maalej and Nabil [22] tested the performance of classic ML algorithms with different feature sets to distinguish bug reports, feature requests, ratings, and user experience. Panichella *et al.* [24] took a similar approach but included slightly broader classes, like information seeking and information giving. Guzmán and Maalej [14] studied how the polarity of sentiment analysis can be applied to classify user feedback.

Automated classification techniques deliver good results in terms of precision and recall, but they are inevitably imperfect. This is largely due to the noise inherently present in user-generated feedback [34], which leads to imperfect classifications that decrease the trust of the user in the algorithm and the platform in which the outputs are embedded [4]. Furthermore, these approaches achieve their best results when using supervised ML algorithms, which require extensive manual work and expensive tagging to train the algorithms. This led to approaches like that of Dhina *et al.* [5], which aims to diminish human effort by employing strategies like active learning. In our work, we want to assess the adequacy of inexpensive crowd workers for the task.

Performing RE activities through crowdsourcing is part of Crowd-based RE (CrowdRE) [13], which includes all approaches that engage a crowd of mostly unknown people to perform RE tasks or provide requirements-relevant information [10]. CrowdRE aims to involve a large number of stakeholders, particularly users, in the specification and evolution of software products. To realize this vision, it is key to understand for which tasks CrowdRE is (cost-)effective. This is one of the goals of our paper, which aligns with the results of Stol and Fitzgerald's [31] case study, which showed that crowdsourcing in software engineering is effective for tasks with low complexity and without interdependencies, such as tagging of user reviews, but less suited for more complex tasks.

3 Kyōryoku: Crowd Annotation for Extracting Requirements-Related Contents from User Reviews

We propose Kyōryoku[2], a method for crowd workers to identify requirements-related contents in online user reviews. In particular, we describe an annotation process that focuses on the separation of useful and useless reviews, and on the identification of reviews that mention requirements-related aspects such as features and qualities. This process can be viewed as a *complex task* (cf. [28]), which crowd workers cannot generally perform because they lack the required level of expertise in RE. For example, laypeople who act as crowd workers are not familiar with the distinction between features and the qualities of these features.

Complex tasks can be outsourced to a large crowd of laypeople by decomposing these tasks into so-called *micro-tasks*; the dominant form of directed crowdsourcing [32]. Micro-tasks involve simpler and more routine data extraction decision workflows that are performed by laypeople in return for relatively small rewards. The difficulty lies in how such a complex task can be structurally transformed into a set of simpler tasks. This involves the definition of effective workflows to guide paid, non-expert workers toward achieving the desired results that are comparable to those that experts would attain [27].

Kyōryoku consists of a stepwise classification process with three phases, as visualized in Figure 1. In line with the *CrowdForge* framework [19], each phase is

[2] Kyōryoku (協力) is a Japanese term for *collaboration*: literally, it combines *strength* (力) with *cooperation* (協).

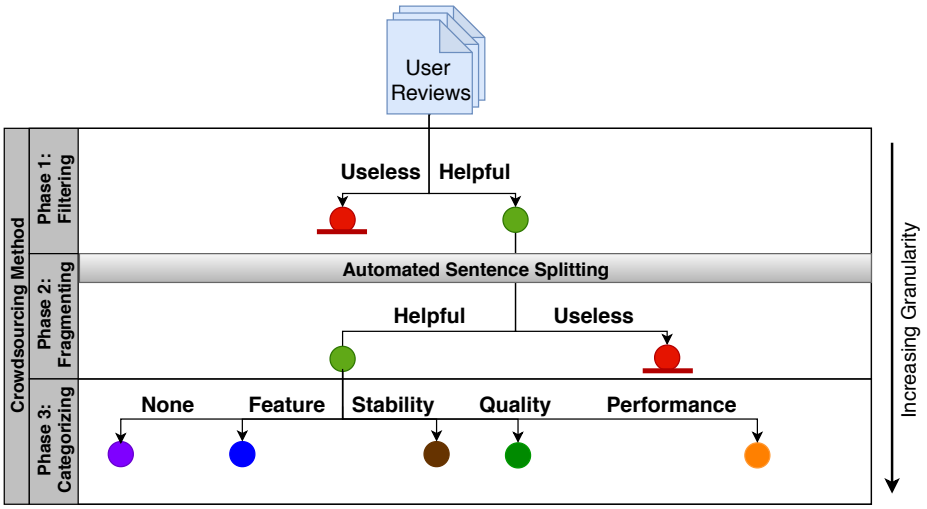

Fig. 1. Overview of the Kyōryoku crowd-based annotation method.

conceived as a micro-task, each being more granular than the preceding phase. Our design approach was iterative and based on empirical evidence: Each micro-task was discussed extensively among three of the authors, and then tested internally with master's degree students in order to maximize the probability that the micro-tasks were defined in a way that they can be executed well by laypeople in a crowdsourcing marketplace. We defined the following phases:

P1: Filter user reviews. Following the principle of increasing granularity [19], crowd workers should first analyze the nature of the data itself before classifying the requirements-relevant aspects. In this phase, crowd workers distinguish between user reviews that are "helpful" to software developers from those that are "useless", i.e., spam or irrelevant. "Spam" is any ineligible user review that is not written with good intent, while a user review is "irrelevant" if it does not contain useful information from an RE perspective. The input for Phase 1 is a set of unprocessed user reviews; crowd workers are presented with the entire body text of each user review.

P2: Filter fragments. Via a text processor, the reviews classified as "helpful" are split into sentences, which we call fragments. The crowd workers perform the same task as in Phase 1, except that they handle one-sentence fragments of helpful user reviews. One-sentence reviews from Phase 1 that are not split up can be kept in the dataset to improve filtering effectiveness.

P3: Categorize. The fragments classified as "helpful" in Phase 2 undergo a more fine-grained classification into five categories. This is a more demanding task for the crowd workers, so it calls for a clear job description with good examples. The category "Feature Request" applies to fragments addressing functional aspects. Three categories are included to denote software quality aspects, and "None of the Above" is used for aspects such as general criticism

and praise. Our categories of software product qualities are based on Glinz' taxonomy [8], which we modified to ease the task and maximize comprehensibility by laypeople. The category "Performance Feedback" reflects the quality "performance". To help crowd workers understand better how "reliability" is distinct, we named it "Stability Feedback", reflecting the reliability aspect most commonly addressed in user feedback [11]. To limit the number of categories, several qualities – including "usability", "portability", and "security" – have been combined into "Quality Feedback".

Our approach emphasizes proper training because crowd workers base the decisions they make during the categorization work on our instructions in the job description. We paid attention to balancing clarity with brevity, both essential properties of a job, i.e., a task assigned to a crowd worker.

Description

In this job, you will be presented with text from user reviews from mobile app stores such as the Google Play Store or the Apple App Store. The goal of this job is to filter out the spam and to remove useless reviews.

Steps

1. Read each user review carefully.
2. Determine whether the review could be of any help to a developer based on the guidelines and examples listed below.
3. Mark the reviews as helpful or useless.

Guidelines

- **Useless Reviews**
  - Contain spam or other unrequested or unwanted messages.
  - Their content does not relate to the app and its functions.
- **Helpful Reviews**
  - Specifically mention aspects of the apps, that is, functions, features and behaviour of an app.
  - Report bugs and performance issues that the user encountered.

Examples

- **Useless Reviews**
  - *"I Really Like This!"*
  - *"My kids love it. Thanks"*
- **Helpful Reviews**
  - *"Newest version crashes when opening"*
  - *"Buggy and unreliable. Does not work often. Signs me out regularly. Won't download movies onto my iPad. Disappointing."*

Fig. 2. Abridged job description for the crowd workers in Phase 1.

Fig. 2 shows an abridged version of the job description for Phase 1, with the template we used for the job description of each phase. All job descriptions are available in our online appendix [33]. The *introduction* triggers the participants' attention, followed by the *steps*, *guidelines*, and *examples*. The guidelines cover the core principles of each answer category, while in the examples, we provide a selection of actual reviews that are representative of these categories. Drafts were tested in two pretests, which showed that the job description required improvements to better guide crowd workers towards the correct decision.

Following the job description, crowd workers are presented with an *eligibility test* that serves two purposes: First, the crowd workers can practice the job, and after the test read the explanations for the items they categorized incorrectly, so they can learn from these mistakes and improve their decision-making. Second, it allows us to ensure that only well-performing crowd workers can participate. The *annotation task* itself is like an eligibility test, with a page presenting a number of items for the crowd workers to categorize.

4 Experiment Design and Conduction

To validate Kyōryoku, we designed a single group experiment for which we recruited crowd workers through the online crowdsourcing marketplace *Figure Eight*[3] to annotate a set of 1,000 user reviews in the three phases shown in Fig. 1. Through our experiment, we sought to confirm the following hypotheses:

H1. Crowd workers can distinguish between useful and useless reviews.
H2. Crowd workers can correctly assign user reviews to different requirement categories.
H3. Extracting RE-relevant contents from online user feedback through crowdsourcing is feasible and cost-effective.

H1 focuses on Phases 1–2, and H2 focuses on Phase 3 of Kyōryoku. H3 is a more general hypothesis regarding the method as a whole.

The *Figure Eight* platform allows crowd workers to perform jobs by assigning micro-tasks in exchange for fixed-price monetary rewards. We set the reward for Phase 1 to $0.03 per user review, based on a pretest in which the participants took an average of 9.3 minutes to classify 50 user reviews. This means that the hourly remuneration is similar to the minimum wage in the United States [15].

In order to test the ability of individual workers to follow Kyōryoku, we decided to offer individual micro-tasks rather than collaborative tasks where crowd workers can assess the contributions of others. However, collaboration and peer reviewing are important research directions to explore in future work.

We opted for an open crowd selection policy: candidates qualify for participation through an eligibility test. We saw no need to add further restrictions such as native language or reputation. Rather, we found it realistic to expect non-native English-speaking crowd workers to be capable of performing such a

[3] https://www.figure-eight.com/

task. If confirmed, this expectation would greatly expand the size of the available crowd and thus the number of crowd workers participating in our micro-tasks.

We selected *Figure Eight* because of its support for data categorization tasks and its many embedded quality control mechanisms, including eligibility test questions defined by the crowdsourcer that crowd workers must pass to contribute, control questions throughout the actual task, and a reputation system.

Our reviews are a sample of Groen *et al.*'s dataset [11]. We omitted the "Smart Products" category, which refers to a combination of hardware and software, and the "Entertainment" category, whose reviews were found not to be representative of the general population of app store reviews in an earlier study [11]. We also discarded the reviews from Amazon's app store, from which reviews can no longer be retrieved, limiting its potential for use in future studies. From the resulting dataset, we took a systematic stratified sample of 1,000 user reviews, in accordance with the job size limit of a *Figure Eight* trial account. The reviews were stratified across apps and app stores, but we limited the proportion of reviews about the Viber app to $\leq 30\%$. The sample resembled the characteristics of the whole dataset with respect to the distribution of stars, sentiment, and years, while the disparity of average app ratings was negligible (maximum +0.16 for TweetCaster Pro).

A gold standard for this dataset was created based on the work of this paper's first author and feedback from other researchers on selected samples. We will compare the crowd work against the gold standard on two different levels of strictness, with the first (*strict*) being the gold standard defined a priori, and the second (*lenient*) being a revision that takes into account potential errors by the researcher, as well as commonly misclassified reviews that can be attributed to ambiguities for which the job description did not provide guidance. The lenient gold standard was constructed after examining the answers by the crowd workers, taking the perspective of the crowd workers, who neither have information regarding the apps to which the reviews refer, nor access to the entire review once it is chunked after Phase 1.

The tags *app reviews*, *spam detection*, and *user reviews* were applied to each test as a means of generating visibility and interest among crowd workers. A total of 45 test questions were constructed to provide 15 unique test questions per phase for quality control purposes. We constructed our test questions to equally represent all possible tagging categories (e.g., all five aspects in Phase 3). For each phase, ten test questions were randomly allocated to the eligibility test. Seven of them had to be answered correctly in order to pass, while the remaining five were used as control questions during the actual task. The workers were presented with pages containing ten items, nine of which were randomly selected fragments of the dataset, and one a control question. A micro-task was limited to five pages, for a total of 50 items, to prevent all the work being done by a small group of early responders. The crowd workers could abandon their job every time they finished a page. We included a setting that disqualified a crowd worker from further participation if they completed a ten-item page too quickly ($< 20s$ for Phases 1 & 2; $< 30s$ for Phase 3). Average time per judgment

varied between 23 (Phase 3) and 14 (Phases 1 & 2) seconds. The test questions along with the job descriptions can be viewed in the online appendix [33].

Table 1. Summary of the configuration of *Figure Eight* per phase.

Phase-Session	Jobs	Judgments per Review	Required Judgments	$ per Judgment
1-1	200	3	600	0.04
1-2	800	3	2,400	0.03
2-1	242	3	726	0.02
2-2	1,000	3	3,000	0.02
3-1	683	6	4,098	0.02

As Table 1 shows, Phases 1 and 2 were carried out in two different sessions. Due to the experimental nature and the limited budget, Phase 1 commenced with a trial job of only 200 reviews to detect possibly overlooked faults or flaws in the process. We were required to split Phase 2 between two accounts because *Figure Eight*'s trial accounts are limited to 1,000 tasks, but we obtained 1,242 fragments from splitting the helpful reviews from Phase 1 into individual sentences. For Phases 1 and 2, three judgments from three different crowd workers were required to reach a satisfactory classification. For Phase 3, we raised this number to six due to the increased complexity of the task with a larger number of categories. Six annotations across five categories moreover precluded a balanced outcome with several categories getting tagged only once. Remuneration varied slightly between the different sessions. Participants in the first session of Phase 1 received a reward that was slightly above average because we had underestimated the efficiency of crowd workers on the platform. Due to budget constraints, Phase 3 offered remunerations slightly below minimum wage for the length of the task.

5 Results

We have organized the results of our experiment as follows: First, we will describe the crowd that we assembled through *Figure Eight* (Sec. 5.1), then present some statistics regarding job duration and cost (Sec. 5.2), and finally report on the outcome of the jobs in terms of precision and recall (Sec. 5.3).

5.1 Demographics of the Gathered Crowd

We gathered a large worldwide crowd through multiple crowd work channels associated with *Figure Eight*. A total of 603 unique crowd workers commenced participation in the five sessions listed in Table 1, 422 of whom passed the eligibility test and quality checks. These 422 workers can be considered contributors. They were from 42 different countries, with the highest number of contributors coming from Venezuela (36.7%), probably due to the current economic situation

[25], followed by Ukraine (11.6%), Russia (7.8%), Egypt (6.6%), and Turkey (6.4%).

An automatically deployed contributor survey showed that the contributors deemed the test questions fair, the tasks not too difficult to complete, and the remuneration satisfactory. However, the overall rating did decrease from 4.3/5 in Phase 1 to 3.7/5 in Phases 2 & 3, probably due to the increasing overall difficulty of the task or the reduced compensation.

The total number of annotations to user reviews or fragments amounted to 10,555. Each contributor classified 24.8 items on average, with the vast majority of crowd workers either tagging the minimum of 10 or the maximum of 50 contributions. On average, 16.4% of the crowd workers failed the eligibility test to perform the job. The failure rates for Phase 1 (9.6%), Phase 2 (11.7%), and Phase 3 (27.4%) highlight the increasing difficulty of the tasks.

5.2 Job Statistics

Table 2 shows that the total cost of our experiment was $354.72, which includes *Figure Eight*'s 20% usage fee. Because we reduced the remuneration due to the tagging of shorter text fragments, Phase 2 was the cheapest. In total, the jobs were active for a total of 323 minutes before reaching full completion. Phase 3 had the highest workload, and therefore took the longest time to reach completion. Phase 2 amassed a large group of contributors the quickest, and therefore achieved completion in the least amount of time.

Table 2. Launch time and completion statistics for all the launched jobs.

Phase-Session	Launch (CET; 2019)	Duration (min.)	Contributions	Test Question Judgments	Total Judgments	Judgments per min.	Time per Judgment	Total Cost ($)
1-1	May 7, 10:47	29	600	477	1,086	38	13.7	33.60
1-2	May 15, 11:27	82	2,400	1,437	3,855	46	12.6	97.20
2-1	May 23, 11:21	28	726	665	1,463	42	12.3	21.36
2-2	May 29, 16:40	44	3,000	2,091	5,250	113	10	84.96
3-1	June 13, 11:04	140	4,098	1,943	6,311	43	23.6	117.60
Total & Micro-Avg		323	10,824	6,613	17,965	56	15.9	354.72

As shown in Fig. 3, the jobs ramped up slowly in the beginning, followed by a period of intense contributions, and finally a long tail for the unfinished jobs to be completed. The average number of judgments per minute was 56, and varied from 38 judgments in session 1-1 to 113 judgments in session 2-2. In the intense contribution phase, which we set as the center 90% contributions in each distribution so as to remove the initial slower phase and the long final tail, the average number of judgments per minute varied from 54 for session 1-1 to 373 for session 2-2. The not-so-steep activity for session 3-1 can be explained by the higher number of contributors failing the eligibility test. The contributors required significantly more time per judgment in Phase 3 (23.6 seconds) than in Phases 1 & 2 (between 10 and 13.7 seconds).

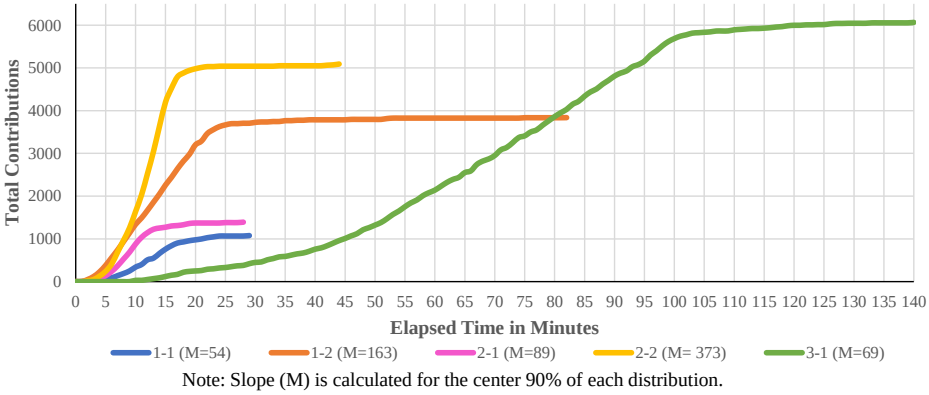

Fig. 3. Total number of contributions received over the course of each session.

5.3 Outcome of the Crowd Work

As summarized in Fig. 4, the crowd workers processed 1,000 reviews from app stores, in which they *identified* 683 requirements-relevant fragments, which they then *classified* into five RE-relevant categories.

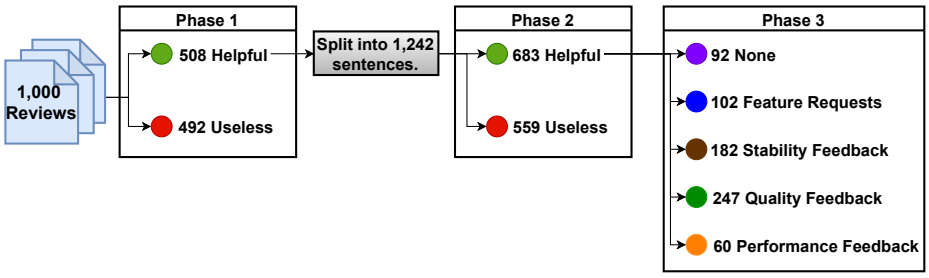

Fig. 4. Overview of the course of the user reviews through the different phases.

Table 3 compares the crowd judgments against the gold standard. In Phase 1, the crowd was able to classify the reviews with a precision of 93%, meaning that only 7% of helpful reviews were misjudged as useless by the crowd. Depending on the strictness of the gold standard (strict or lenient, see Sec. 4), the crowd was able to correctly identify either 74% or 84% of the useless reviews from the dataset (recall)[4].

In Phase 2, the crowd was able to classify useless results with a precision of 88%, meaning that 12% of helpful fragments were discarded incorrectly. The

[4] Note: because Phases 1 and 2 focus on filtering out irrelevant reviews, we take the *useless* category as our positives.

Table 3. Detailed comparison of the results of the crowd for Phases 1 & 2.

Phase	Positives: Useless (Gold Std)	Negatives: Helpful (Gold Std)	True Positives	True Negatives	False Positives	False Negatives	Precision	Recall
1 (strict)	620	380	459	347	33	161	0.93	0.74
1 (lenient)	547	453	460	421	32	87	0.93	0.84
2 (strict)	679	563	478	482	81	201	0.86	0.70
2 (lenient)	609	633	493	566	66	117	0.88	0.81

crowd was able to identify 81% of all useless fragments. This constitutes effective filtering, although 19% of the useless fragments still remained in the dataset. Fragments that received the same judgments from all three contributors (61.4% of all cases) were more often classified correctly, reaching 87% accuracy, while accuracy dropped to 64% for the cases in which only two contributors agreed.

As Table 4 shows, the crowd workers reached an average accuracy of 78% (lenient) in Phase 3. The confusion matrix in Table 5 reveals that there was some misalignment between categories, mainly between "None" and "Quality" and, to a lesser degree, "Performance". Crowd workers were the most precise in classifying "Stability" issues, while reaching the highest recall on "Feature Requests". They were the least precise on "None" and "Performance" issues, and reached the lowest recall on "Performance" issues. Further investigation of the agreement between the six contributors per review fragment (Table 4) revealed a meaningful impact of the level of agreement on the accuracy of the classification. Accuracy ranged from 100% for fragments that the six contributors classified unanimously, down to 49% when only two contributors picked the same category.

Table 4. Accuracy for the different levels of agreement between contributors.

Agreement	Frequency	Correct	Incorrect	Accuracy
Six out of six	85 (12%)	85	0	100%
Five out of six	144 (21%)	131	13	91%
Four out of six	170 (25%)	145	25	85%
Three out of six	196 (29%)	128	68	65%
Two out of six	88 (13%)	43	45	49%
Total	**683 (100%)**	**532**	**151**	**78%**

6 Threats to Validity

Despite our efforts to carefully design Kyōryoku, not every aspect could be accounted for due to the experimental nature of this research. Kyōryoku relies on several assumptions due to the scarcity of literature on how to assemble effective micro-tasks. Thus, we have no way of knowing whether Kyōryoku reached its highest or lowest potential, which makes it harder to put the results into context.

Table 5. Confusion matrix for the results of the crowd in Phase 3 (lenient).

		Gold Standard						
		None	Feature	Stability	Performance	Quality	Precision	Recall
Crowd	None	67	5	3	3	14	0.57	0.73
	Feature	4	94	1	1	2	0.83	0.92
	Stability	14	8	134	6	20	0.93	0.80
	Performance	4	5	3	29	19	0.63	0.41
	Quality	28	1	3	7	208	0.80	0.84
	Average						0.75	0.74

Furthermore, it is currently impossible to trace back potential flaws to individual design decisions, because only one such experiment was conducted and no variations have been tested so far. However, the effectiveness of the training method seems to be the most crucial aspect, due to the high number of participants who failed the eligibility test.

The tests were conducted with only limited experience with the *Figure Eight* platform, and without prior experience in outsourcing tasks to the crowd. No restrictions were in place to exclude countries or channels that might provide results with significantly lower quality. The analysis of the results currently does not account or compensate for possible influences from these sources. Comparing the results against the gold standard, however, did not reveal significant discrepancies for any particular country or channel in terms of accuracy.

Each phase of our experiment utilized inputs from the preceding phases; thus, errors by crowd workers were perpetuated in all subsequent phases. Although this affected the cumulative results, we decided to examine the performance of the whole method performed sequentially, not that of individual phases. Testing each phase separately might lead to slightly different results. An inherent shortcoming of this approach is furthermore that the classifications are left to a very small subset of the crowd, with only three judgments in Phases 1 & 2, and six in Phase 3. As Table 4 corroborates, the quality of the results can be improved by involving more crowd workers, although this also increases costs. Finally, the dataset contains user reviews from 2011–2015; due to the rapid evolution of the app landscape, results may differ with more recent user feedback.

The creation of the gold standard and the review of the crowdsourcing task's outputs relied mostly on a single researcher, with other researchers cross-checking samples. Thus, although we transparently share our materials publicly, only samples of the gold standard classification have been reviewed. It is not unreasonable to assume that some errors were introduced into the gold standard that may affect the validity of the results.

Finally, the quality control mechanisms that we deployed into the *Figure Eight* platform have an effect on the results, for they determine the inclusion or exclusion of crowd workers. Despite our efforts to make it as robust as possible, this quality control mechanism is imperfect. This might especially affect the potential accuracy by incorrectly excluding good workers or by improperly detecting poor workers.

7 Conclusion & Future Work

We have presented Kyōryoku, a crowdsourcing method for identifying and classifying user requirements – more precisely, requirements-relevant information – in online user feedback through crowd work. Kyōryoku was tested on 1,000 app store reviews, which were analyzed and classified by over 400 crowd workers.

Based on the outcomes of Phases 1 & 2 of Kyōryoku, we can confidently state that crowd workers are able to distinguish between useful and useless reviews (**H1**). The crowd workers achieved precision rates of 93% and 88% and recall rates of 84% and 81%, respectively, in these phases. Although there is no automated technique that serves as a baseline, further research is needed to compare against algorithms based on automated spam detection in app reviews [2].

When we consider the ability of the crowd workers to correctly assign user reviews to different requirement categories (**H2**), the results are positive, but inevitably not as good as the binary useful/useless classification. The overall accuracy was 74% for the five categories that we deemed suitable for crowdsourced classification: "Feature", "Stability", "Performance", (other) "Quality", and "None". Interestingly, for the 85 fragments with perfect agreement among all six taggers, we could observe 100% accuracy. We have not tested Kyōryoku against automated classifiers yet. These results seem to be at least as good as optimized automated classifiers of NFRs [21], which achieve an accuracy of ∼70%.

H3 concerned the feasibility and cost-effectiveness of Kyōryoku to extract RE-relevant contents from online user feedback. We were able to show the feasibility of such a method through the tasks we composed for the crowd workers to carry out. In terms of cost-effectiveness, 1,000 reviews were fully processed through crowdsourcing for approximately $350 and in 5.4 hours for all phases and sessions combined. On the other hand, creating a gold standard, i.e., tagging the data without crowd workers, required circa 20–30 person-hours. Although we cannot provide a conclusive answer to H3, the results suggest that Kyōryoku might be suitable for companies who wish to analyze user reviews about their products, but who do not have sufficient resources to hire an expert assessor.

This work presents a novel method for engaging a crowd to elicit user requirements from online user feedback, and paves the way for future work in this direction. Kyōryoku, which includes openly available task descriptions [33], can be taken as is and used by organizations who would like to classify a reviews dataset. Kyōryoku can be improved by changing the wording of the job description, the examples, and the classification taxonomy. To do so, it is imperative to complement our quantitative results with a qualitative analysis that reveals which utterances are most likely to lead to false positives and false negatives. We hope that future studies will take Kyōryoku as a baseline to improve upon; researchers can directly compare their automated or human-driven method using the gold standard we make available. Alternatively, it is possible to use this gold standard to train approaches based on ML. Also, it would be interesting to investigate whether the crowd can effectively use fine-grained taxonomies of quality requirements. It is essential to test the approach on larger datasets that contain recent user reviews. The outcomes of such an analysis might also have financial

consequences: Is Kyōryoku feasible for companies whose products receive thousands of user reviews per day? Moreover, different aggregation techniques can be studied to reconcile the taggers' opinions. Finally, in the context of adopting crowdsourcing for analyzing large-scale industrial datasets requires assessing the ethical concerns [6] that crowd work entails, since most contributors originate from countries with a complex social and political situation.

More generally, this research advocates the use of crowdsourcing for *complex tasks* in RE or other disciplines. Our results warrant increased exploration of the applicability of crowdsourcing to similar challenges that revolve around large volumes of data with a difficult nature. This research has shown that crowd workers are able to deal with perhaps more complex problems than anticipated, provided they receive proper instruction.

References

1. Castro-Herrera, C., Duan, C., Cleland-Huang, J., Mobasher, B.: Using data mining and recommender systems to facilitate large-scale, open, and inclusive requirements elicitation processes. In: Proc. of RE, pp. 165–168 (2008)
2. Chandy, R., Gu, H.: Identifying spam in the iOS app store. In: Proc. of WebQuality, pp. 56–59 (2012)
3. Chen, N., Lin, J., Hoi, S.C., Xiao, X., Zhang, B.: AR-miner: Mining informative reviews for developers from mobile app marketplace. In: Proc. of ICSE, pp. 767–778 (2014)
4. Dalpiaz, F., Parente, M.: RE-SWOT: From user feedback to requirements via competitor analysis. In: Proc. of REFSQ, pp. 55–70 (2019)
5. Dhinakaran, V.T., Pulle, R., Ajmeri, N., Murukannaiah, P.K.: App review analysis via active learning. In: Proc. of RE, pp. 170–181 (2018)
6. Fort, K., Adda, G., Cohen, K.B.: Amazon mechanical turk: Gold mine or coal mine? Computational Linguistics **37**(2), 413–420 (2011)
7. Gadiraju, U., Fetahu, B., Kawase, R.: Training workers for improving performance in crowdsourcing microtasks. In: Design for Teaching and Learning in a Networked World, pp. 100–114. Springer (2015)
8. Glinz, M.: On non-functional requirements. In: Proc. of RE, pp. 21–26 (2007)
9. Glinz, M.: A glossary of requirements engineering terminology. Version 1.7. International Requirements Engineering Board (IREB). Available at https://www.ireb.org/en/cpre/cpre-glossary/ (2017)
10. Glinz, M.: CrowdRE: Achievements, opportunities and pitfalls. In: Proc. of CrowdRE, pp. 172–173 (2019)
11. Groen, E.C., Kopczyńska, S., Hauer, M.P., Krafft, T.D., Doerr, J.: Users—The hidden software product quality experts? a study on how app users report quality aspects in online reviews. In: Proc. of RE, pp. 80–89 (2017)
12. Groen, E.C., Schowalter, J., Kocpzyńska, S., Polst, S., Alvani, S.: Is there really a need for using NLP to elicit requirements? A benchmarking study to assess scalability of manual analysis. In: Proc. of NLP4RE (2018)
13. Groen, E.C., Seyff, N., Ali, R., Dalpiaz, F., Doerr, J., Guzman, E., Hosseini, M., Marco, J., Oriol, M., Perini, A., Stade, M.: The crowd in requirements engineering: The landscape and challenges. IEEE software **34**(2), 44–52 (2017)

14. Guzman, E., Maalej, W.: How do users like this feature? A fine grained sentiment analysis of app reviews. In: Proc. of RE, pp. 153–162 (2014)
15. Horton, J.J., Chilton, L.B.: The labor economics of paid crowdsourcing. In: Proc. of EC, pp. 209–218 (2010)
16. Hosseini, M., Groen, E.C., Shahri, A., Ali, R.: CRAFT: A crowd-annotated feedback technique. In: Proc. of CrowdRE, pp. 170–175 (2017)
17. Hosseini, M., Phalp, K.T., Taylor, J., Ali, R.: Towards crowdsourcing for requirements engineering. In: Proceedings of REFSQ Workshops (2014)
18. Howe, J.: The rise of crowdsourcing. Wired **14**(6), 1–4 (2006)
19. Kittur, A., Smus, B., Khamkar, S., Kraut, R.E.: CrowdForge: Crowdsourcing complex work. In: Proc. of UIST, pp. 43–52 (2011)
20. Lim, S.L., Finkelstein, A.: StakeRare: Using social networks and collaborative filtering for large-scale requirements elicitation. IEEE Transactions on Software Engineering **38**(3), 707–735 (2012)
21. Lu, M., Liang, P.: Automatic classification of non-functional requirements from augmented app user reviews. In: Proc. of EASE, pp. 344–353 (2017)
22. Maalej, W., Nabil, H.: Bug report, feature request, or simply praise? On automatically classifying app reviews. In: Proc. of RE, pp. 116–125 (2015)
23. Nayebi, M., Cho, H., Ruhe, G.: App store mining is not enough for app improvement. Empirical Software Engineering **23**(5), 2764–2794 (2018)
24. Panichella, S., Di Sorbo, A., Guzman, E., Visaggio, C.A., Canfora, G., Gall, H.C.: How can I improve my app? Classifying user reviews for software maintenance and evolution. In: Proc. of ICSME, pp. 281–290 (2015)
25. Posch, L., Bleier, A., Flöck, F., Strohmaier, M.: Characterizing the global crowd workforce: A cross-country comparison of crowdworker demographics. arXiv preprint arXiv:1812.05948 (2018)
26. Renzel, D., Behrendt, M., Klamma, R., Jarke, M.: Requirements Bazaar: Social requirements engineering for community-driven innovation. In: Proc. of RE, pp. 326–327 (2013)
27. Retelny, D., Robaszkiewicz, S., To, A., Lasecki, W.S., Patel, J., Rahmati, N., Doshi, T., Valentine, M., Bernstein, M.S.: Expert crowdsourcing with flash teams. In: Proc. of UIST, pp. 75–85 (2014)
28. Schenk, E., Guittard, C.: Towards a characterization of crowdsourcing practices. Journal of Innovation Economics & Management **1**(7), 93–107 (2011)
29. Snijders, R., Dalpiaz, F., Brinkkemper, S., Hosseini, M., Ali, R., Ozum, A.: REfine: A gamified platform for participatory requirements engineering. In: Proc. of CrowdRE, pp. 1–6 (2015)
30. Stanik, C., Haering, M., Maalej, W.: Classifying multilingual user feedback using traditional machine learning and deep learning. In: Proc. of AIRE (2019)
31. Stol, K.J., Fitzgerald, B.: Two's company, three's a crowd: A case study of crowdsourcing software development. In: Proc. of ICSE, pp. 187–198 (2014)
32. Valentine, M.A., Retelny, D., To, A., Rahmati, N., Doshi, T., Bernstein, M.S.: Flash organizations: Crowdsourcing complex work by structuring crowds as organizations. In: Proc. of CHI, pp. 3523–3537 (2017)
33. van Vliet, M., Groen, E., Dalpiaz, F., Brinkkemper, S.: Crowd-annotation results: Identifying and classifying user requirements in online feedback (2020), https://doi.org/10.5281/zenodo.3754721, Zenodo
34. Williams, G., Mahmoud, A.: Mining Twitter feeds for software user requirements. In: Proc. of RE, pp. 1–10 (2017)