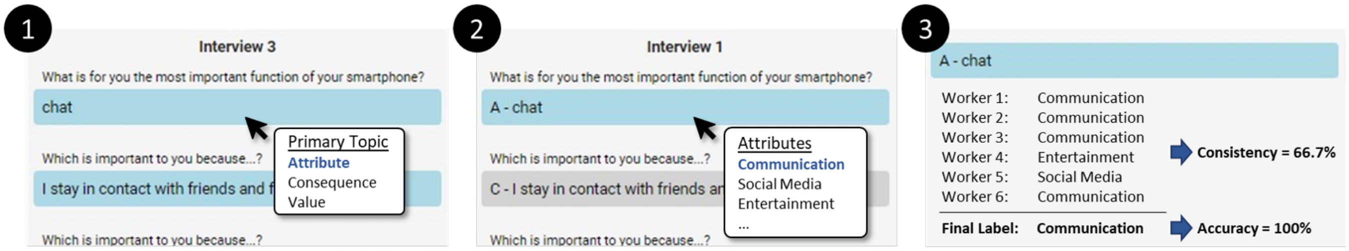
Accelerating Deductive Coding of Qualitative Data: An Experimental Study on the Applicability of Crowdsourcing

Saskia Haug
saskia.haug@kit.edu
Karlsruhe Institute of Technology
Karlsruhe, Germany

Tim Rietz
tim.rietz@kit.edu
Karlsruhe Institute of Technology
Karlsruhe, Germany

Alexander Maedche
alexander.maedche@kit.edu
Karlsruhe Institute of Technology
Karlsruhe, Germany



What is for you the most important function of your smartphone?
chat
Which is important to you because...?
I stay in contact with friends and f
Which is important to you because...?

What is for you the most important function of your smartphone?
A - chat
Which is important to you because...?
C - I stay in contact with friends ar
Which is important to you because...?

A - chat

Worker 1:	Communication
Worker 2:	Communication
Worker 3:	Communication
Worker 4:	Entertainment
Worker 5:	Social Media
Worker 6:	Communication
Final Label:	**Communication**

Consistency = 66.7%
Accuracy = 100%

Figure 1: Our interactive coding system enables a crowd of non-experts to code semi-structured qualitative data. (1) First, the workers code the primary topics of all interview answers. (2) Then, the workers code each interview answer with the respective specific codes in separate tasks for each primary topic. (3) Finally, the workers' most mentioned code is set as final label. The agreement among workers indicates the crowds' consistency and the agreement between the workers' and experts' final label shows the accuracy of the crowd.

ABSTRACT

While qualitative research can produce a rich understanding of peoples' mind, it requires an essential and strenuous data annotation process known as coding. Coding can be repetitive and time-consuming, particularly for large datasets. Crowdsourcing provides flexible access to workers all around the world, however, researchers remain doubtful about its applicability for coding. In this study, we present an interactive coding system to support crowdsourced deductive coding of semi-structured qualitative data. Through an empirical evaluation on Amazon Mechanical Turk, we assess both the quality and the reliability of crowd-support for coding. Our results show that non-expert coders provide reliable results using our system. The crowd reached a substantial agreement of up to 91% with the coding provided by experts. Our results indicate that crowdsourced coding is an applicable strategy for accelerating a strenuous task. Additionally, we present implications of crowdsourcing to reduce biases in the interpretation of qualitative data.

CCS CONCEPTS

• **Human-centered computing** → *Interactive systems and tools*;
• **Information systems** → **Crowdsourcing**.

KEYWORDS

Crowdsourcing, Coding, Qualitative Data, Empirical Evaluation

ACM Reference Format:
Saskia Haug, Tim Rietz, and Alexander Maedche. 2021. Accelerating Deductive Coding of Qualitative Data: An Experimental Study on the Applicability of Crowdsourcing. In *Mensch und Computer 2021 (MuC '21), September 5–8, 2021, Ingolstadt, Germany.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3473856.3473873

1 INTRODUCTION

Collecting and analyzing qualitative data allows researchers to gain a deep understanding of peoples' opinions, thoughts and feelings. Often, qualitative data is collected through interviews following a semi-structured approach. To draw conclusions from qualitative data, researchers must annotate each data item with a short code or label. This process is called coding and is a fundamental step during qualitative data analysis (QDA). After the identification of initial categories the codes are revised and assigned to data items to gain a nuanced understanding of topics and themes. In *deductive coding* a set of predefined codes is applied to the dataset, while in *inductive coding* the codes are developed during coding [50, 59]. Researchers value the process of qualitative coding for the rich insights it provides. Unfortunately, coding is a time-consuming and tedious task that becomes prohibitive for large datasets [47]. Additionally, the coder's perspective on the data might bias the coding results [22]. Although the coders' perspective on the data always introduces some bias, interpretation is an essential element of most QDA methods. Diverse perspectives that result from personal experiences and the social, cultural and economical background of the coder are even appreciated in the HCI and CSCW communities [51].

MuC '21, September 5–8, 2021, Ingolstadt, Germany
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8645-6/21/09…$15.00
https://doi.org/10.1145/3473856.3473873

Due to the infinite potential dimensions of textual data [4], the coding process is difficult to automate. Various algorithms have been developed to support semi-automated deductive coding processes with human-defined rules and natural language processing (NLP) [9, 47, 64] or machine learning (ML) [17, 43, 64], yet researchers lack trust in coding support based on artificial intelligence (AI) [38, 47].

Meanwhile, crowdsourcing is a nascent approach for tasks involving large amounts of data, like the labeling of images [6, 70]. However, research on applying crowdsourcing to accelerate qualitative coding is sparse [38]. Crowdsourcing not only provides the opportunity of outsourcing tedious tasks without much effort, but it further offers access to an enormous number of workers spread all over the world. Their diversity can be of high value for increasing the objectivity of data analyses. So far, researchers across disciplines remain doubtful about the applicability of crowdsourcing due to the lack of evidence showcasing the reliability and quality of crowdsourced results [38, 62, 66]. In particular, recent studies faced difficulties in achieving a high quality of labels in crowdsourcing [1] which shows that a suitable design of the crowdsourcing task is necessary.

This paper addresses the applicability of crowdsourcing as an approach for scaling deductive coding of semi-structured qualitative data. Building on recent work of the HCI community, we introduce an interactive coding system. We evaluate the proposed system with semi-structured data in the form of interviews following the hard-laddering approach. In hard-laddering, participants produce answer that link concrete attributes with consequences and with abstract values that they seek to achieve. Participants produce these chains in hard-laddering one-by-one [53]. As such, in the context of this study, we understand semi-structured data as interview data that follows a fixed questioning structure, while the number and content of participants' answers are varying. We present the results of our experimental study with two goals: Firstly, we aim to evaluate the quality of workers' labels by understanding *if workers' codes differ from experts' codes and what might be the causes for that?* Secondly, we aim to assess the workers' reliability by determining *if workers agree with each other and what affects this agreement?* The term 'experts' refers to researchers who are experienced in qualitative coding as they previously applied QDA methods in research projects. We evaluate the system by having 170 workers code 240 qualitative interviews on Amazon Mechanical Turk (MTurk) and analyze the reliability of workers' codes as well as their agreement with experts' codes. In this paper we present:

- A conceptualization and implementation of an interactive coding system for coding semi-structured qualitative data enabling non-expert crowdworkers to apply a codebook to semi-structured interview data. The system includes suitable methods and designs for ensuring a high quality coding process.
- A summative evaluation comparing crowdworkers' coding with experts' coding, showing that workers can code reliably and achieve a substantial agreement with experts' codes when using our system and process.

Our novel contributions include the development of a system for crowdsourcing codes that is specially designed for semi-structured qualitative data and adapts the complexity of deductive coding to workers' abilities by splitting up the task (Figure 1). Compared to traditional coding systems, like Atlas.ti and MaxQDA, and new approaches based on AI, like search-query style code rules [47] or Cody [64], our system minimizes the effort for expert coders as the workload of coding is transferred from one to several shoulders. This can help to reduce the bias of coding results as at least four individual opinions on the correct code are incorporated in the creation of the final code. Depending on the category, crowdworkers achieve the same performance as experts using our system, in terms of established intercoder reliability measures. With an increasing abstractness of the codes, the performance of both workers and experts worsens. With our work, we provide an approach for accelerating the deductive coding of semi-structured qualitative data and provide empirical support for the applicability of crowdsourcing for QDA. Further, we discuss differences between experts and workers in the usage behavior of codes; implications for applying the coding system; limitations of our work; and avenues for future research.

2 RELATED WORK

In the following section, we give a brief summary of current research on semi-automated coding of qualitative data. This is followed by an introduction into crowdsourcing with a special focus on coding tasks.

2.1 Coding in QDA

Some form of coding, where researchers assign codes or labels to small chunks of text, is involved in most QDA approaches. The codes are usually single words or short paragraphs and reflect information about the content of the piece of text [19, 25, 61, 71]. *Inductive* and *data-driven coding* means researchers develop emergent codes while reading and analyzing the data multiple times [50, 59]. Codes then represent ideas and trends that are found in the data. *Deductive* and *theory-based coding* means codes are founded in theories or hypothesis and are defined a priori [50, 59]. While inductive coding enables a higher accuracy and completeness, deductive coding is superior for achieving a higher precision [45], thus a higher agreement among coders. The reliability of codes is usually expressed by the intercoder agreement which indicates the agreement between multiple coders who coded the dataset independently [46]. Due to its iterative and creative characteristics and the infinite potential dimensions of textual data, the coding process is considered time-consuming and error-prone, even for experienced coders [47]. For organizing, retrieving, coding and analyzing qualitative data, QDA software, like Atlas.ti, MaxQDA and Nvivo, has established. However, these systems still provide only limited support for automating the coding process [47]. Lu and Shulman [44] developed a coding analysis toolkit that allows users to perform the coding task using only keystrokes. They estimate the coding task to be two to three times faster with their tool than with common QDA software. Although this toolkit provides a high reduction of time, it still requires a qualified expert to spend their time on coding the whole dataset. More promising approaches apply AI. The systems based on ML algorithms are either trained separately [43] or learn and improve their models while the user is coding the actual dataset [64]. Cody, a system to semi-automate coding [64] allows users to define and adjust code rules that Cody uses to make code suggestions. Additionally, Cody uses supervised ML to extend previous manual coding to seen and unseen data. Aeonium is a ML-based system that is able to predict ambiguous data on which the codes of two independent coders will not agree [17]. The ML model is again trained on prior coding decisions of both coders. Approaches using NLP apply human-created rules to automate coding [9, 41, 47]. However, researchers still lack trust in AI-based coding support as they assume computers not being able to apply a human-like interpretation of the data [47]. Therefore, all AI-based coding systems still need researchers to code at least a small subset of the data as it is preferred by researchers [47]. Crowston et al. [10] compared both AI-based approaches and identified drawbacks for both of them: While NLP rules still require an expert to develop the rules, ML-based approaches need many coded examples to learn from. Additionally, the coding results are in both cases biased by the perspective of one expert. This paper suggests an approach that neither requires the involvement of any kind of experts, nor needs a large dataset to train algorithms. By crowdsourcing the codes, multiple diverse perspectives are considered and compiled to one final label.

2.2 Crowdsourcing

In crowdsourcing, tasks that are usually performed by humans with domain expertise are outsourced to a large undefined group of people, the so-called *workers* [29]. Thanks to the power of the crowd, the workers are able to complete these tasks without the expertise that was initially required. The tasks are usually published by individuals or organizations, the so-called *requesters*, on third-party platforms like MTurk. In exchange for the completion of the task, workers receive a payment that considers the complexity of the task and the estimated completion time [13]. For overcoming the lack of expertise, complex tasks are often split into multiple easier tasks [48, 58, 73]. Additionally, tasks usually consist out of multiple *assignments*, whereby each assignment stands for one worker doing the task. The submissions can then be combined to the final result. MTurk is a prominent crowdsourcing platform that allows requesters to implement individual user interface designs for task execution. Besides the user interface design, the task setup is known to have a high influence on the attractiveness of the task [6, 15, 24, 39, 73]. The attractiveness is important to attract more workers and influences how fast the task is completed. However, it has been shown that it has no influence on the quality of results [11, 49, 74]. The task setup includes the task title, keywords, the payment, filters for worker qualifications, the number of assignments and the duration of the task. The average duration workers need to complete one task can vary between a few seconds [6] and almost one hour [48]. One downside of crowdsourcing is the unpredictable reliability of workers. Malicious behaviour of workers reduces the reliability and quality of the results and consequently poses a significant obstacle for task requesters [23]. Besides malicious workers, there are also workers who are unintentionally producing low quality work, because they are not focusing on the task [36]. The effort needed for checking all results is comparable to the time and cost of doing the task itself [30]. There are multiple approaches besides the traditional filtering to check the quality and reliability of results. The approaches can be classified in four different groups: (1) Approaches that are based on comparing individual worker input with the consensus of the crowd [30, 54] or their own input [23], (2) approaches that utilize other workers for validating submissions [6, 16, 27], (3) approaches that compare selected worker input with a predefined gold-standard [40, 54, 57], and (4) approaches that use the worker's behaviour for assessing their reliability [65]. As shown by Mitra et al. [54] a combination of methods for increasing the quality can be beneficial. Quality controls vary greatly in terms of costs, implementation effort, and results. Methods that work for one use case well might have the opposite effect on another task [28].

Two important measurements for data quality are consistency and accuracy [69]. In our study these values represent reliability and quality of codes as indicated in Figure 2. Consistency indicates if a data value is the same in the same situation [69]. In our context, consistency refers to the agreement among workers and therefore represents the reliability of results. The accuracy is defined by "the closeness to the true value characteristic of such measurements" [18], which is in our case the agreement between the crowd code and the expert code. The accuracy determines the quality of our results. While the accuracy of the crowd as a measure for the quality of results is applied in multiple similar studies [6, 16, 31, 67], the consistency for evaluating the reliability of results is applied rather less [51]. We believe that considering the consistency of workers when analyzing the quality of results might bring additional insights into the applicability of crowdsourcing. When developing crowdsourcing tasks, various aspects of the task design and setup must be considered. Especially the reliability and quality of results is a big concern of requesters [23, 36]. Considering the individual requirements of a task is crucial for selecting an appropriate task design and setup and thereby ensuring a high reliability and quality of results. Unlike related studies, we pay attention to the consistency of workers by analyzing the agreement among workers.

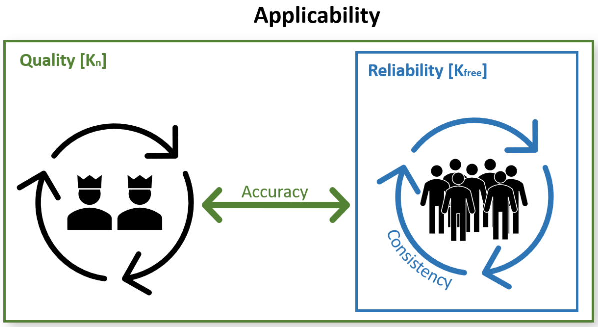


Figure 2: In our study, the applicability of crowdsourcing is defined by the reliability and quality of workers' codes. The reliability is indicated by the workers' consistency, which describes the agreement among workers. The quality of the results is specified by the crowds' accuracy. Therefore, the agreement between the aggregated experts' codes and the aggregated workers' codes is calculated.

2.3 Coding as a Crowdsourcing Task

The related work contains multiple studies on crowdsourced coding or labeling of diverse data types, such as multimedia data in the form of videos [37, 55, 68] or pictures [6, 12]. Crowdsourcing labels for multimedia data is a common method, especially for the training of supervised ML algorithms. In the simplest cases, users must provide in these tasks only a "Yes/No/Maybe" selection [6, 13, 70]. For this specific context, different approaches for aggregating and validating individual labels were suggested [6]. Research on the coding of textual data, especially semi-structured qualitative data, is limited [2, 31, 72]. A common assumption for crowdsourcing the coding of textual data is that every text has one correct code that is recoverable by the consensus of the crowd [3]. However, recent studies highlight the diversity of the crowd and appreciate disagreements in coding as indications of ambiguities that hint at problems with the codebook or alternative interpretations of the source information [6, 16, 33]. Often, the coding of text data is only one step of a more complex task. For example, users in the study of Marge et al. [48], had to rate the importance of utterances of transcribed meeting speeches to create good summaries. André et al. [2] applied crowdsourcing to identify Wikipedia barnstar categories. Users had to group the barnstars and assign names to the clusters. This task is an example of crowdsourcing inductive coding as no categories were predefined. Specifically, they discovered that providing context for coding increases the quality of results. Irvine et al. [31] aim to create entity recognition models based on crowdsourced annotations of an e-mail dataset. Users must label persons, organizations, and locations in e-mails and distinguish between named and unnamed entities. The models that were trained with the gathered data compare favorably to models trained with expert annotations. Wilson et al. [72] produced similarly promising results in their study on annotating privacy policies. Hence, prior work indicates that non-expert workers are able to produce high-quality labels with an appropriate task design.

Although there is already much research on coding text data, it lacks investigating the coding of qualitative textual data with a certain structure. When coding qualitative interviews, the particular answer is necessary for finding the most appropriate code, but also preceding and subsequent answers need to be considered. Therefore, a suitable user interface is required that enables workers to understand the context of interview answers. Additionally, it needs to be examined if non-expert workers can consider the context when coding interview answers. None of the previous studies have either provided an approach for coding semi-structured qualitative data or developed a user interface that might be adaptable to the coding of qualitative interview data. Still, the presented work shows promising results for harnessing the diversity of the crowd for coding. Therefore, in this study, we present a system design that can achieve a reliability and quality of crowdsourced codes comparable to the ones of experts, proving that crowdsourcing is applicable for coding semi-structured qualitative data. Additionally, we highlight the advantages of considering multiple diverse perspectives in the coding process and discuss the applicability of our coding system during different steps of the coding process. For instance, different perspectives can be helpful to evaluate the ambiguity of the codebook and sharpen codes even before the actual coding process.

3 DESIGNING AN INTERACTIVE CODING SYSTEM FOR CROWDSOURCING DEDUCTIVE CODING

In the following, we describe the interactive coding system that was developed in an iterative process. The coding system presents semi-structured qualitative data and a codebook. It enables non-expert users to code the presented data items with the codes of the provided codebook. Figure 3 shows the interface of the coding system. The system design can be split in three subdesigns, with each having its own requirements: (1) the user interface design, (2) the setup of the task on the crowdsourcing platform, including aspects like payment, task title and the filtering, and (3) the postprocessing of the crowdsourced results. The system requirements and the resulting design will be explained in the following.

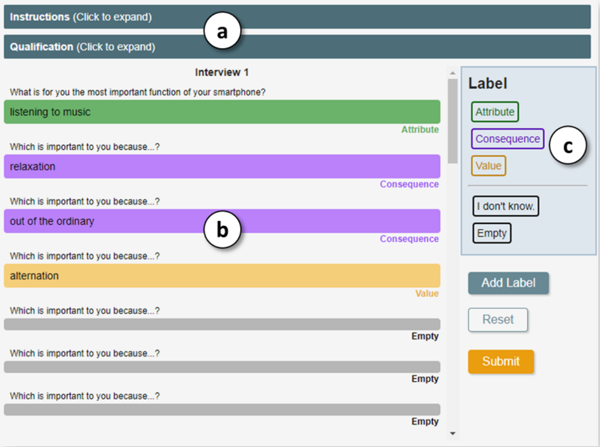
3.1 System Requirements

We began this study with an exploratory literature review in the ACM digital library searching for different combinations of the terms "crowd*", "cod*", "label*", "automat*" "qualitative data", "QDA" and "interview". By analyzing the results we derived five requirements for a coding system for non-expert workers that consider the essential challenges of designing crowdsourcing tasks. While this list might not necessarily be conclusive, it summarizes the requirements most frequently and prominently included in the related work. The requirements specifically focus on the needs of qualitative researchers and serve as a basis for our system's design.

- *R1 Adaptation of Task Complexity:* The coding of qualitative data is usually done by experts with years of experience [47]. To enable non-experts who never coded textual data before to understand the requirements of the task, the task complexity shall be adapted to the abilities of non-experts.
- *R2 Provision of Codebook:* Qualitative researchers want to interact with their data and create a codebook before they outsource the deductive coding of their data [47]. Therefore, the system shall provide a codebook to workers that was developed by the requester. Thereby, researchers can provide basic guidance for workers but workers' interpretation of the data is still required to identify the most suitable code.
- *R3 Contextualize Coding:* Providing context and showing multiple items at once not only improves the quality of coding results [2], but also increases the speed of coding [34]. The system shall provide workers context for coding the data.
- *R4 Enforce Reliability and Quality:* The low reliability and quality of results is a main reason for potential requesters to avoid crowdsourcing [30, 36, 38]. In order that crowdsourcing codes achieves a reduction of workload for the qualitative researchers, the system must validate the trustworthiness of the results. The system needs to include means to ensure a high reliability and quality of results. This includes identifying and excluding malicious workers when necessary.
- *R5 Attractiveness of Task Design and Setup:* Workers freely choose which tasks they want to work on. Tasks with a low attractiveness due to a low payment or other factors are consequently not or only slowly processed by workers [11, 49, 74]. Therefore, the task setup shall have an attractive design, including title, payment and task complexity.

3.2 UI Design for Crowdsourcing Deductive Coding

We aimed to keep the user interface design as simple as possible to support workers in completing the task efficiently (R5) (Figure 3). The interface shows multiple interviews underneath each other (R3). The exact number of interviews displayed in one task should depend on the number of answers per interview and the number of codes to not overload the worker. In our evaluation the total number of answers to be coded varied between four and 28 in four to ten interviews (Figure 4). The codebook contains between five and nine codes. The unit of analysis is set to one interview answer, represented as box, to simplify the coding and aggregation process. Longer interview answers can still be split into multiple boxes when preparing the data. Next to the interviews, the codebook is displayed (R2). This codebook includes codes and a short description for each code. Short descriptions of the codes shall ensure that users understand what each code means. They are displayed when hovering over a code. For empty or unclear answers, the two codes 'Empty' and 'I don't know' are added to the codebook. Before starting the task, the users are required to read instructions and take a qualification test to check whether they have understood the instructions correctly (R4). The test consists of several interview answers, of which at least 60%[1] must be coded correctly. If 60% is not reached, the user has to restart from the beginning. This is an established approach to ensure that malicious workers are not able to participate in the task, and thereby harm the quality of results [54]. Interview answers can be coded by selecting one or multiple answer boxes and exactly one code in the codebook and then clicking on *Add Label*. The workers' process is visualized in Figure 4.



Interview 1

What is for you the most important function of your smartphone?

listening to music

Attribute

Which is important to you because...?

relaxation

Consequence

Which is important to you because...?

out of the ordinary

Consequence

Which is important to you because...?

alternation

Value

Which is important to you because...?

Empty

Which is important to you because...?

Empty

Which is important to you because...?

Empty

Figure 3: The design of the user interface of the interactive coding system includes (a) two separate boxes for the instructions and the qualification test, (b) multiple interviews underneath each other, and (c) a codebook. Here, all answers are already coded.

3.3 Task Design and Setup for Crowdsourcing Deductive Coding

The design of the task as it is perceived by users is not only determined by the UI design but also by the task setup. A common approach to reduce the task complexity and the cognitive load for users is dividing complex tasks into multiple less complex tasks (R1). We verified in several pre-tests that splitting the task leads to significantly better coding results. In literature it is recommended to apply around 20 codes that can be clustered into five to seven themes, regardless of the size of the database [8]. This facilitates splitting up the process. First, the primary topic to which an interview response belongs can be coded. Based on the results of the first step, the final codes can then be assigned in separate tasks for each primary topic (Figure 4). This reduces the number of codes that can be used in one task remarkably. While the UI design remains the same for both steps, the task setup can vary due to different estimated durations and levels of complexity.

The probably most important aspect of the task setup is the payment (R5). We discovered in our pre-tests that bonus payments lead to better results than a fixed payment, which is consistent with findings of recent studies [39, 56]. Therefore, the fixed payment was reduced to 0.04$ and a completion bonus and a quality bonus were implemented. The completion bonus is paid when the worker codes all interview answers and the quality bonus is paid for each code that matches the code of at least one other worker on the task. Incentivizing workers to match the consensus of the crowd is known to increase the quality of results [54] (R4). While an agreement between two workers does not necessarily indicate a "correct" answer, it provides coders with an incentive not to submit rash codings. However, this approach might disadvantage workers with very unique interpretations.

The maximum possible payment was stated in the task title along with the expected duration. This was intended to create maximum transparency for users, as it allows them to directly weigh up the effort and benefits (R5). Additionally, this has shown to reduce the dropout rate of workers [6]. For ensuring that no spammers are allowed to the task, the task setup includes a filtering for an acceptance rate of over 95% (R5). No other filters were applied to ensure that a large and diverse crowd can work on the task. As Snow et al. [67] discovered that four non-experts are able to replace one expert annotator, we collected four codes per answer in our pre-tests. It showed that often one of the four coders produced a lower quality of codes than the others and that often one coder does not code all displayed interview answers. Consequently, for the final evaluation each task was conducted by six workers so that at least four codes per answer are collected (R5).

3.4 Postprocessing of Results

For aggregating the resulting codes the coding system must be able to identify low quality workers and exclude their submissions (R5). Additionally, the collected codes must be aggregated to one final code. As workers are incentivized to match the consensus of the crowd, workers who do not match at least one worker in multiple cases are likely to have an overall low quality of results.

[1] 60% is an arbitrary value that we derived as optimal for our use case as part of formative user-testing of the system. However, for other use cases, a different value might provide better results.

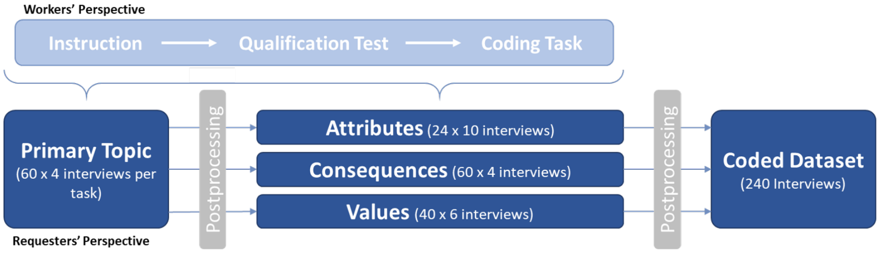


Figure 4: The process of the crowdsourcing task from the workers' perspective (top) and from the requesters' perspective (bottom). It is for each type of task indicated how many interviews are included in one task.

Therefore, the system calculates for each submission the percentage of submitted codes that agree with at least one other worker on the task. If this percentage is too low, the submission is excluded from the determination of the final code. Finding the optimal threshold is one of the goals of the summative evaluation. The final code is determined by the majority of the workers' codes. In case there is no majority, the requester must make the final decision.

4 EVALUATION

We evaluated the developed coding system through an empirical study on MTurk. The evaluation aimed to investigate if workers are able to achieve similar codes to experts by analyzing the workers accuracy and consistency. In this section the study design is presented, followed by the results of the evaluation.

4.1 Crowdsourcing Platform

MTurk was chosen as crowdsourcing platform for the evaluation because of its flexibility, its size, the topics of tasks and the previously conducted research. MTurk is a popular crowdsourcing platform that allows requesters to design and develop tasks completely independent and adapted to individual requirements. With around 500.000 workers from over 190 countries, MTurk is not only a huge but also a very diverse crowdsourcing platform. Workers are mainly from India and the United States, which ensures a high number of workers available at every time of the day [20]. An international crowd also provides more diversity regarding the knowledge and expertise of workers. There are many tasks on MTurk related to the interpretation and analysis of data, like categorization and classification tasks [15]. Workers should therefore be experienced in coding tasks. The last reason for choosing MTurk is that most of previous research focused on designing tasks for MTurk. Therefore, best-practices and other insights are fully applicable to our task design.

4.2 Study Design

For assessing the reliability of workers' codes, we first focus on the agreement among workers, the consistency. The consistency can be measured as consensus of the workers to select the most suitable code for a given data item [3]. Additionally, we determine the accuracy of results by comparing workers' codes with the codes defined by experts. This serves to understand to what extent the results of experts and the crowd differ. For the evaluation of the coding system, the dataset from Rietz and Maedche [63] was taken. They investigated the usage of chatbots for performing laddering interviews and replicated the research study "What a smartphone is to me" by Jung [32] using chatbot technology. Therefore, participants of the interviews were asked for their favorite smartphone apps and the underlying reasons why they favor these apps. The interviews were conducted with students of a German university and were subsequently translated to English to make them usable on an international crowdsourcing platform like MTurk. Outsourcing the coding of the data from their study is especially interesting as it would show that not only the data acquisition but also the data analysis does not need the involvement of qualified experts and can be outsourced to an anonymous crowd. For laddering interviews the decision on the two coding steps is less complex than for other qualitative interviews. During laddering interviews, following a means-end theory [26], interviewees are asked a series of "why is that important to you?" questions in order to progress from simple product attributes, via the consequences of using the product, to the underlying values of the user [53]. Consequently, the primary topics for laddering interviews are already set to attributes, consequences and values. In the second step the specific codes will be assigned in separate tasks for each group (Figure 4).

The codebook for the second step was developed and iterated by two of the authors of this paper following established iterative processes. In an iterative process, they coded a subset of the data, compared their results and refined the codebook until they achieved a satisfying intercoder agreement [46]. For the primary topics they stopped at an intercoder agreement of 0.81 K_n and for the specific codes at 0.73 K_n. K_n is a modification of Cohen's Kappa for free-marginal coding [5]. Free-marginal coding means that the number of data items per code is not defined before. An intercoder agreement of 0 means the agreement is similar to chance agreement, while an intercoder agreement of 1 indicates perfect agreement [7]. The final codebook for specific codes contains 20 codes (six for attributes, eight for consequences, six for values) that are explained by a short description. For determining the experts' codes, the same two authors first coded the primary topics, agreed then on a final topic for each answer and finally coded with the specific codes. They achieved an intercoder agreement of 0.81 K_n for the primary topics and 0.74 K_n for the specific codes. All disagreements of specific codes had to be discussed until an agreement was found.

Table 1: Agreement between workers (consistency) for each type of tasks and agreement between the two expert coders. 'Codes' means the number of possible codes without 'I don't know'.

		Crowd without Threshold		Crowd with Threshold		Experts	
	Codes	Percentage	K_{free}	Percentage	K_{free}	Percentage	K_n
Primary Topic	4	88.8%	0.65	89.9%	0.68	93.3%	0.81
Attributes	6	89.4%	0.78	89.4%	0.78	98.4%	0.96
Consequences	8	70.9%	0.49	71.6%	0.50	85.4%	0.67
Values	6	70.3%	0.46	73.3%	0.52	87.6%	0.70

4.3 Results

For the evaluation, we compare the results of the interactive coding system using crowdsourcing with the codes determined by experts and examine the reliability and quality of the results by calculating the consistency and accuracy of the crowdworkers. The evaluation consisted out of 240 interviews that were split into four batches of each 60 interviews. By doing so, we could post the batches separately and increase the reliability of our evaluation by ensuring that our results are independent of the time of the day or the availability of specific workers. The evaluation was divided according to the two coding steps. First, the primary topics were coded and processed. Then, based on the results of the first step, attributes, consequences, and values were successively coded with the specific codes and processed (Figure 4).

The consistency of the results is a common means for evaluating the reliability of coding results and is determined by the agreement between the workers. For this, we calculated K_{free}, which is a modification of Cohen's Kappa for free-marginal multi-rater coding [60]. K_{free} is therefore an adaption of K_n for comparing codes among more than two individual coders. Additionally, we determined the average percentage of workers that agreed on the most mentioned code of an answer. The difference between these two measures is, that only K_{free} considers the number of different codes mentioned by workers. However, the percentage is a more tangible and comprehensible value. The accuracy is measured by the percentage agreement between worker codes and expert codes and the respective K_n [5]. During the main evaluation, 1132 assignments were considered. These assignments were completed by 139 unique workers who submitted up to 133 assignments each, whereby the most diligent 20% of the workers completed 80% of the tasks. We noticed that many workers had excessive completion times with over one hour. When having a closer look, we observed that these workers first accepted multiple assignments before they started to work on them. Consequently, the completion time of MTurk is not considered suitable for analysis.

In the following analysis the codes for empty answers were not included as identifying empty answers is very simple and not the focus of the evaluation. The 'I don't know' codes are treated as the same as all other codes. For determining the final primary topics of answers for being able to proceed with coding with specific codes, an optimal threshold was found at 82% which was a compromise between the target of four submissions per answer and a high level of agreement between workers' and experts' codes. Consequently, all submissions in which less than 82% of the codes match the codes in at least one other submission were excluded. We also experimented with excluding unreliable workers for the following tasks with a similar quality check. Therefore, we searched for the highest accuracy for each type of task. The optimal threshold was found at 0% for attributes, 33% or less for consequences, and 51-59% for values. Consequently, a threshold for attributes and consequences is not necessary as it does not increase the accuracy of workers' codes and even for values, the increase of the accuracy is only 0.5%. Without excluding unreliable workers, the consistency of workers ranges between 0.46 and 0.78 K_{free}, and the average percentage agreement of workers on the final code is always between 70% and 90% depending on the type of task (Table 1). On average 79.9% of workers agree on the final code. With the quality threshold, this value can be increased to 81.0%. When coding the primary topics and the attributes, workers' agreement is much better than when coding consequences and values. These findings are consistent with the agreement between the two experts. They had for all types of codes a higher agreement among each other than workers. However, the percentage agreement for experts considers only one or two experts, thus it can be either 50% or 100% of experts, mentioning the final code, while for workers the lowest possible percentage agreement is 16.7%, meaning all six workers mentioned a different code. Furthermore, K_{free} and K_n are calculated differently and are therefore not comparable. Still, the experts' consistency shows the same phenomenon as workers' codes as it is much higher for the primary topic and attributes than for consequences and values. According to Landis and Koch [35] the workers' results indicate a moderate agreement for consequences and values and a substantial agreement for attributes and the primary topic. The Pearson's correlation coefficient shows for the workers' results a significant negative correlation of -0.36 between the number of codes (without 'Empty' and 'I don't know') and the workers consistency with a confidence interval of 95%.

In total, the merged workers' codes agree with experts' codes with and without a quality threshold in 65.7% which is a K_n of 0.64 (Table 2). When considering only codes, where workers and experts agreed on the primary topic, the average agreement is 72.6% or a K_n of 0.69. Both values indicate a substantial agreement between workers and experts [35]. The accuracy varies significantly according to the type of code, with the lowest agreement being 0.44 K_n for values and the highest and almost perfect agreement being 0.89 K_n for attributes [35]. The results of all four batches do not show any significant outliers which proves that our results are not a onetime success and the evaluation results are reliable.

When comparing our results on the accuracy to similar approaches in existing research, the number of coders as well as the number of different codes need to be considered. Additionally,

Table 2: Agreement between workers and experts (accuracy) for each primary topic and each batch without a quality check. 'Codes' means the number of possible codes without 'I don't know'.

	Codes	Batch 1	Batch 2	Batch 3	Batch 4	Average	K_n
Total	20	65.6%	65.4%	70.0%	62.0%	65.7%	0.64
Primary Topic	3	89.7%	92.0%	93.9%	86.5%	90.5%	0.86
Specific	-	73.1%	71.1%	74.6%	71.1%	72.6%	0.69
Attributes	6	86.9%	95.0%	90.5%	90.5%	90.7%	0.89
Consequences	8	60.3%	59.2%	64.2%	56.0%	60.0%	0.56
Values	6	52.2%	49.0%	61.9%	47.5%	52.0%	0.44

it should be recognized that different Kappa values are not comparable with each other. Furniss and Blandford [21] achieved a K_{free} of 0.48 when they let five non-experts label tweets with eight codes. At the same time, McMinn et al. [52] achieved an intercoder agreement of 0.81 when letting at least three workers label tweets with eight event categories. However, both studies considered three to five coders for each data item, while we included four to six worker codes for each answer. Additionally, in both studies coders had longer texts with more information to base their coding decisions on than in our experiment. Our results for the total agreement between workers and expert lies with 0.64 exactly in the middle of these two values, while our results for primary topics and attributes are even better than the results of McMinn et al. [52]. Considering the mentioned influencing factors, we achieved with our proposed design an extremely good agreement between worker and expert codes compared to existing studies. Although only half of all values are coded by workers equally as by experts, most of the workers (on average 70.3%) agreed on the final code. A similar effect is achieved for consequences. This demonstrates that the aggregated worker codes, even if they do not agree with experts' codes, are reliable and valid.

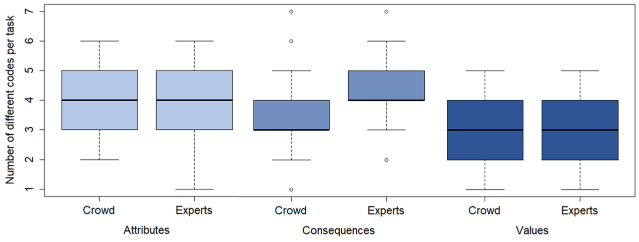


Figure 5: The number of different codes per task for workers and experts.

Comparing the final workers' codes with the experts' codes, we noticed that workers tend to switch their codes for consequences less often than experts. When coding consequences, workers used on average 3.5 different codes (SD = 1.20) per task, while experts used 4.3 different codes (SD = 1.05) for the same interviews (Figure 5). When looking at the diagram, one must take into account that more codes were offered for coding consequences (eight) than for attributes (six) and values (six). Nevertheless, a comparable number of different codes was used for all three types of tasks. A further observation when taking a closer look at the coding of consequences is that experts and workers have very different frequencies of use of the individual codes (Figure 6). Some codes were used by workers more frequently than by experts, while others are only rarely used. For example, workers used the code *Simplification of (Physical) Tasks* only nine times, while experts used it over 80 times on the same dataset. It also shows that for some codes, experts were confident about, workers had a strong agreement on a different code. An example of this is the attribute *WhatsApp* which we coded as *Communication* while workers coded it as *Social Media*.

Besides the quantitative analysis, we also considered worker feedback that we received on the tasks. While some workers sent e-mails when they accidentally submitted the task too early or had questions about the task, the qualification, or the bonus payment, some workers sent direct feedback with their opinion about the payment and some features. One worker stated they "*always like to do [our] task at any where & any time*". This worker asked for our batch schedule and the upcoming tasks and was willing to set an alarm at 4 am local time to work on our tasks.

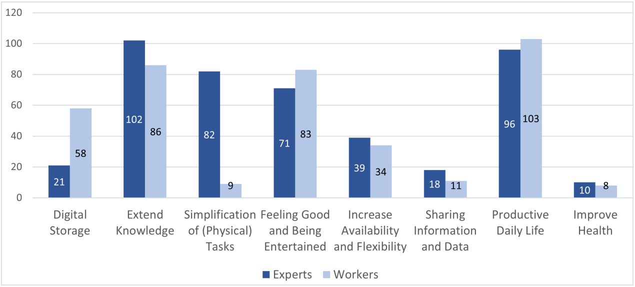


Figure 6: The utilization of individual codes for consequences for workers and experts.

5 DISCUSSION

This study provides empirical evidence of the applicability of crowdsourcing codes for qualitative data analysis of semi-structured data. We showed that a substantial quality and reliability of codes is achievable with novice coders in a crowdsourced setting. Overall, both quality and reliability are highly dependent on the abstractness of the codes. In the following, we discuss important implications and limitations of our results.

5.1 Varying Consistency and Accuracy

Both consistency and accuracy achieved the best results for attributes and the worst results for values, respectively. We believe three aspects cause the varying results for consistency and accuracy: Firstly, we already showed that the workers' consistency negatively correlates with the number of possible codes. The coding task becomes increasingly complex for crowdworkers with a larger codebook. However, as our codebook contains the same amount of codes for consequences and values, the number of codes does not explain the differences in agreement for these two categories. Secondly, we expect accuracy and consistency for attributes and primary topics to be higher due to their lower abstractness of codes. In our evaluation context, attributes usually refer to apps or smartphone features that everyone knows and has used before, while consequences and especially values refer to abstract ideas and personal experiences. The accuracy of deductive coding strongly depends on the domain familiarity of workers [45]. While attributes only require being familiar with essential smartphone apps, consequences and values require putting oneself in the interviewee's place and understanding their usage behavior and underlying needs. Both authors, who coded the dataset as experts, have a similar social, cultural and economical background as the majority of the interviewees which might impact their interpretation of the data. For workers, that do not know the interviewees and might have a very different personal background, considering the interviewees' perspective might be challenging. For interview topics that are very subject-specific and require special knowledge, this effect might be even bigger and requires recruiting a crowd that has this specific knowledge. Also, the concept of attributes, consequences, and values might be new to workers. Thirdly, attributes are easier to detect and distinguish from consequences and values. Consequences and values are regularly mistaken for one another. When the primary topic is wrongly coded in the first place, the specific codes do not fit, leaving workers confused by the available choices.

Over 70% of workers agreed on the final codes for consequences and values. Further, high consistency of workers in combination with a low accuracy might indicate that the workers found another possible code for the interview answer. When considering Bachrach et al. [3], who state that the crowd's consensus can find the correct label, it could even mean that personal experiences or expectations might bias the experts' code and might be a worse choice than the crowd consensus. However, we recommend researchers appreciate the diverse codes and see them as an encouragement to revise the codebook. For instance, a section that crowdworkers uniformly interpret with another code than experts might indicate a flaw in the experts coding, or a simpler or alternative interpretation.

5.2 Differences in the Usage Behaviour of Codes

Workers tend to switch codes less often than experts when coding consequences. We assume that our task design might not encourage workers to think about the best code for each interview answer individually. Some crowdworkers might attempt to finish tasks rapidly by supplying ill-fitting responses [23]. Thus, in our case, workers sometimes apply one code for multiple interview answers at once, even if another code might fit better. Compared to attributes and values, the tasks for consequences often included consecutive consequences in an interview. Usually, consecutive consequences cover a similar topic but should not necessarily be coded with the same code. Still, this setup might have tempted workers to save time by coding multiple answers the same. Additionally, by doing so, workers do not have to spend time understanding all available codes. The workers usually used a maximum of five to six codes, potentially indicating that the cognitive load of working with larger codebooks became too high. Experts did not use the coding tool and had to think about each answer individually. As experts would consider their agreement with their respective co-coder, time savings by applying the same code to multiple answers despite of the content was offset by the necessary clean-up work as part of discussing codings with co-coders. Furthermore, experts were familiar with all possible codes and did not have to spend additional time understanding them. These drawbacks of the UI design and the codebook design need to be considered when using the coding system. However, we aimed to address this issue through the quality bonus payment included in our incentivization structure. By disclosing to coders that we would pay a bonus for agreements between coders, we directed coders to behave more similar to the experts in our setup through incentivizing considering the own agreement with co-coders. Although this might disadvantage workers with unique perspectives on the data by granting them a lower bonus, this is a necessary feature for avoiding random coding. To avoid the related ethical issue and further encourage workers to share their personal interpretations, additional quality measures need to be developed and tested in future studies.

Besides the lower number of different codes in a task, workers also use the codes very differently than experts (Figure 6). The differences might be caused by different usage scenarios depending on the personal experiences of the workers. This is an excellent example to showcase the influence of diverse perspectives on the coding results. Individual interpretations of particular codes and answers must be taken into account when using crowdsourcing. When appropriately managed, diverse interpretations can significantly benefit the requester and the final codes' reliability and quality.

5.3 Applying the Coding System in Practice

We developed an interactive coding system that leads to high-quality codes by relying on the crowd's consensus. This system is designed especially for researchers and analysts who appreciate diverse perspectives and are interested in heterogeneous interpretations of their data. The HCI and CSCW communities, in particular, make frequent use of qualitative research methods and appreciate diversity with regards to the interpretation of results [51]. Our coding system's main application is the coding of large datasets with a predefined codebook (deductive coding). We guide the post-processing of results by recommending a specific threshold for the agreement between workers to be considered reliable and for the respective coding to be included in the aggregated code. Thereby, we assist researchers and analysts in evaluating workers' codings in cases with low consistency. We found that for attributes and consequences, a quality threshold is not beneficial for increasing the accuracy. Consequently, we assume that the rest of the crowd in these cases compensated malicious or inattentive workers' submissions. This might indicate that publishing six assignments per task is a good choice for our coding system. Still, it might be interesting to analyze the effects of varying the number of assignments per task on the crowd's consistency and accuracy. In our study, we assumed that only one correct code exists for each answer. However, for other use cases with less strict requirements, it might be useful to allow multiple codes to be correct and adapt the aggregation and analysis of the codes accordingly.

Besides the initial usage scenario, researchers might also profit in other contexts from the advantages of crowdsourcing codes. We identified three additional use cases: Firstly, the coding system could be applied to test an early version of a codebook. The workers' diverse perspectives on the explanations in the codebook and on the dataset might help researchers identify unclear or confusing codes. Secondly, the crowd could replace or serve as a proxy for the second expert coder. Such a replacement would still significantly lower the effort for the experts while reducing potential bias introduced by the experts' perspectives. Nevertheless, the expert retains control over the coding process and can choose the final code considering the workers' codes. We encourage researchers to interpret inconsistencies in crowdsourced codes, not as noise but as valuable heterogeneous data interpretations and a potential sign to improve the codebook further.

5.4 Limitations and Future Work

We demonstrated the presented interactive coding system's potential for achieving agreement on crowdsourced deductive coding of semi-structured qualitative data. As our system is a rather novel and controversial approach for deductive coding, we assume that further research is essential to convince researchers of the advantages of crowdsourcing codes. Therefore, in this section, we present the limitations of our study and provide suggestions for future research. Firstly, our study results might not be generalizable for other contexts than the use case presented. We evaluated our system with laddering interviews that offer many advantages over other semi-structured interviews, such as the standardized sequence of questions, the relatively short and focused answers, and the predefined primary topics for answers. Interviews with less structure or longer and more verbose answers might have problems fitting in the system design. Thus, there is value in evaluating the applicability of the presented coding system for other types of semi-structured qualitative data. Additionally, we are aware that in practice qualitative researchers often apply significantly more than 20 codes in total. As we assume workers' accuracy and consistency being highly dependent on the number of codes, future research could investigate how many codes can be better handled by workers.

Furthermore, we did not analyze the influence of the individual design features on the reliability and quality of results. While some aspects might have a significant impact on the outcomes, others might not necessarily be relevant. An example is the UI design, which might encourage workers to use fewer codes during a task. Future work should focus on analyzing the impact of specific design features on outcome variables like consistency, accuracy, the average duration of assignments, and the perceived attractiveness of the task. This way, more general design knowledge about developing crowdsourcing tasks for qualitative data analysis can be built up to foster scalable user research.

Another limitation of our work is the lack of data on the participating workers' demographics. As there already exist multiple studies that investigate workers' demographics and validate their diversity (e.g. [14, 42]), we decided for our study to focus on the general applicability of crowdsourcing for deductive coding. Future researchers could investigate the effect of specific social, cultural and economical backgrounds of workers on their perspective and interpretation of qualitative data. This research could contribute a better understanding of potential impacts on the quality of coding and external influences on codes. In this context, it could also be interesting to examine the effect of the relationship between the worker and the interviewee. When all coders participated in the same laddering interview, their understanding of the topic and the related consequences and values could be higher, which might lead to a better quality of results. As the scalable collection of qualitative data produces another key challenge to include wide audiences in qualitative research, a combination of our system with automated elicitation systems (e.g., Ladderbot [63]) might provide a comprehensive end-to-end solution for qualitative researchers.

Finally, the relatively low consistency of workers for consequences and values might be an indication for ambiguities in our codebook. A codebook that was iteratively refined with the help of the crowd could have led to even better results. Future research should investigate how inconsistencies in crowdsourced codes could best be instrumentalized to revise the codebook. In this context, approaches for expert-worker collaboration should also be taken into account.

6 CONCLUSION

In this work, we presented a new approach for scaling the deductive coding of qualitative interview data using crowdsourcing. Applying the use case of coding laddering interviews as an example of semi-structured qualitative data, we design an interactive coding system that was deployed on MTurk. We review best practices for MTurk and principles for successful crowdwork task design and present five system requirements for crowd-based coding systems. In an empirical study, we demonstrate the applicability of crowdsourced deductive coding using the proposed interactive coding system. MTurk workers' coding with our system achieved a commendable agreement with experts' coding. However, the accuracy and consistency depend on the interview answers' complexity and the abstractness of codes in the codebook. With this work, we show that crowdsourcing for deductive coding is an applicable approach for accelerating the qualitative data analysis.

REFERENCES

[1] Omar Alonso, Catherine C Marshall, and Marc Najork. 2013. Are Some Tweets More Interesting Than Others? #HardQuestion. In *Proceedings of the Symposium on Human-Computer Interaction and Information Retrieval (HCIR '13)*. Association for Computing Machinery, New York, NY, USA, 1–10. https://doi.org/10.1145/2528394.2528396

[2] Paul André, Aniket Kittur, and Steven P Dow. 2014. Crowd Synthesis: Extracting Categories and Clusters from Complex Data. In *CSCW '14*, Susan Fussell, Wayne Lutters, Meredith Ringel Morris, and Madhu Reddy (Eds.). Association for Computing Machinery, New York, NY, USA, 989–998. https://doi.org/10.1145/2531602.2531653

[3] Yoram Bachrach, Tom Minka, John Guiver, and Thore Graepel. 2012. How to Grade a Test without Knowing the Answers: A Bayesian Graphical Model for Adaptive Crowdsourcing and Aptitude Testing. In *Proceedings of the 29th*

International Coference on International Conference on Machine Learning (ICML'12). Omnipress, Madison, WI, USA, 819–826.

[4] Avrim L. Blum and Pat Langley. 1997. Selection of Relevant Features and Examples in Machine Learning. *Artificial Intelligence* 97, 1-2 (1997), 245–271. https://doi.org/10.1016/s0004-3702(97)00063-5

[5] Robert L. Brennan and Dale J. Prediger. 1981. Coefficient Kappa: Some Uses, Misuses, and Alternatives. *Educational and Psychological Measurement* 41, 3 (1981), 687–699. https://doi.org/10.1177/001316448104100307

[6] Joseph Chee Chang, Saleema Amershi, and Ece Kamar. 2017. Revolt: Collaborative Crowdsourcing for Labeling Machine Learning Datasets. In *CHI '17*, Gloria Mark, Susan Fussell, Cliff Lampe, M. C. Schraefel, Juan Pablo Hourcade, Caroline Appert, and Daniel Wigdor (Eds.). Association for Computing Machinery, New York, NY, USA, 2334–2346. https://doi.org/10.1145/3025453.3026044

[7] Jacob Cohen. 1960. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement* 20, 1 (1960), 37–46. https://doi.org/10.1177/001316446002000104

[8] John Creswell. 2016. *30 Essential Skills for the Qualitative Researcher* (1. ed.). SAGE Publications, Inc., Thousand Oaks, CA, USA.

[9] Kevin Crowston, Eileen E Allen, and Robert Heckman. 2012. Using Natural Language Processing Technology for Qualitative Data Analysis. *International Journal of Social Research Methodology* 15, 6 (2012), 523–543. https://doi.org/10.1080/13645579.2011.625764

[10] Kevin Crowston, Xiaozhong Liu, and Eileen E. Allen. 2010. Machine learning and rule-based automated coding of qualitative data. *Proceedings of the American Society for Information Science and Technology* 47, 1 (11 2010), 1–2. https://doi.org/10.1002/meet.14504701328

[11] Florian Daniel, Pavel Kucherbaev, Cinzia Cappiello, Boualem Benatallah, and Mohammad Allahbakhsh. 2018. Quality Control in Crowdsourcing: A Survey of Quality Attributes, Assessment Techniques, and Assurance Actions. *Comput. Surveys* 51, 1 (2018), 1–40. https://doi.org/10.1145/3148148

[12] Jia Deng, Olga Russakovsky, Jonathan Krause, Michael S Bernstein, Alex Berg, and Li Fei-Fei. 2014. Scalable Multi-Label Annotation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Matt Jones, Philippe Palanque, Albrecht Schmidt, and Tovi Grossman (Eds.). Association for Computing Machinery, New York, NY, USA, 3099–3102. https://doi.org/10.1145/2556288.2557011

[13] Xuefei Nancy Deng, K. D. Joshi, and Robert D. Galliers. 2016. The Duality of Empowerment and Marginalization in Microtask Crowdsourcing: Giving Voice to the Less Powerful Through Value Sensitive Design. *MIS Quarterly: Management Information Systems* 40, 2 (2016), 279–302. https://doi.org/10.25300/MISQ/2016/40.2.01

[14] Djellel Difallah, Elena Filatova, and Panos Ipeirotis. 2018. Demographics and Dynamics of Mechanical Turk Workers. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, Vol. 9. Association for Computing Machinery, New York, NY, USA, 135–143. https://doi.org/10.1145/3159652.3159661

[15] Djellel Eddine Difallah, Michele Catasta, Gianluca Demartini, Panagiotis G. Ipeirotis, and Philippe Cudré-Mauroux. 2015. The Dynamics of Micro-Task Crowdsourcing: The Case of Amazon MTurk. In *WWW 2015 - Proceedings of the 24th International Conference on World Wide Web*. Association for Computing Machinery, New York, NY, USA, 238–247. https://doi.org/10.1145/2736277.2741685

[16] Ryan Drapeau, Lydia B Chilton, and Daniel S Weld. 2016. MicroTalk: Using Argumentation to Improve Crowdsourcing Accuracy. In *Hcomp*. AAAI Publications, Palo Alto, CA, USA, 32–41. https://www.aaai.org/ocs/index.php/HCOMP/HCOMP16/paper/viewFile/14024/13630

[17] Margaret Drouhard, Nan Chen Chen, Jina Suh, Rafal Kocielnik, Vanessa Pena-Araya, Keting Cen, Xiangyi Zheng, and Cecilia R. Aragon. 2017. Aeonium: Visual analytics to support collaborative qualitative coding. In *2017 IEEE Pacific Visualization Symposium (PacificVis)*. IEEE, New York, NY, USA, 220–229. https://doi.org/10.1109/PACIFICVIS.2017.8031598

[18] Churchill Eisenhart. 1968. Expression of the Uncertainties of Final Results. *Science* 160, 3833 (1968), 1201–1204. https://doi.org/10.1126/science.160.3833.1201

[19] Jeanine C Evers. 2018. Current issues in qualitative data analysis software (QDAS): A user and developer perspective. *Qualitative Report* 23, 13 (2018), 61–73. https://nsuworks.nova.edu/tqr/vol23/iss13/5

[20] Karen Fort, Gilles Adda, and Kevin Bretonnel Cohen. 2011. Amazon Mechanical Turk: Gold Mine or Coal Mine? *Computational Linguistics* 37, 2 (2011), 413–420. https://doi.org/10.1162/COLI

[21] Dominic Furniss, Jonathan Back, and Ann Blandford. 2012. Cognitive resilience: can we use Twitter to make strategies more tangible?. In *Proceedings of the 30th European Conference on Cognitive Ergonomics*. Association for Computing Machinery, New York, NY, USA, 96–99. https://doi.org/10.1145/2448136.2448156

[22] Patricia Fusch and Lawrence Ness. 2015. Are We There Yet? Data Saturation in Qualitative Research. *Qualitative Report* 20 (2015), 1408–1416.

[23] Ujwal Gadiraju, Ricardo Kawase, Stefan Dietze, and Gianluca Demartini. 2015. Understanding Malicious Behavior in Crowdsourcing Platforms: The Case of Online Surveys. In *Conference on Human Factors in Computing Systems - Proceedings*, Bo Begole, Jinwoo Kim, Kori Inkpen, and Woontack Woo (Eds.), Vol. 2015-April. Association for Computing Machinery, New York, NY, USA, 1631–1640. https://doi.org/10.1145/2702123.2702443

[24] Catherine Grady and Matthew Lease. 2010. Crowdsourcing Document Relevance Assessment with Mechanical Turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk (CSLDAMT '10)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 172–179.

[25] Justin Grimmer and Brandon M. Stewart. 2013. Text as data: The promise and pitfalls of automatic content analysis methods for political texts. *Political Analysis* 21, 3 (2013), 267–297. https://doi.org/10.1093/pan/mps028

[26] Jonathan Gutman. 1982. A Means-End Chain Model Based on Consumer Categorization Processes. *Journal of Marketing* 46, 2 (1982), 60. https://doi.org/10.2307/3203341

[27] Derek L Hansen, Patrick J Schone, Douglas Corey, Matthew Reid, and Jake Gehring. 2013. Quality control mechanisms for crowdsourcing. In *CSCW 2013*, Amy Bruckman, Scott Counts, Cliff Lampe, and Loren Terveen (Eds.). Association for Computing Machinery, New York, NY, USA, 649. https://doi.org/10.1145/2441776.2441848

[28] Chien Ju Ho, Aleksandrs Slivkins, Siddharth Suri, and Jennifer Wortman Vaughan. 2015. Incentivizing High Quality Crowdwork. In *WWW 2015 - Proceedings of the 24th International Conference on World Wide Web*. Association for Computing Machinery, New York, NY, USA, 419–429. https://doi.org/10.1145/2736277.2741102

[29] Jeff Howe. 2008. *Crowdsourcing: Why the Power of the Crowd Is Driving the Future of Business* (1 ed.). Crown Publishing Group, New York, NY, USA.

[30] Panagiotis G. Ipeirotis, Foster Provost, and Jing Wang. 2010. Quality Management on Amazon Mechanical Turk. In *Workshop Proceedings - Human Computation Workshop 2010, HCOMP2010*. Association for Computing Machinery, New York, NY, USA, 64–67. https://doi.org/10.1145/1837885.1837906

[31] Ann Irvine and Alexandre Klementiev. 2010. Using Mechanical Turk to Annotate Lexicons for Less Commonly Used Languages. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk (CSLDAMT '10)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 108–113.

[32] Yoonhyuk Jung. 2014. What a Smartphone is to Me: Understanding User Values in Using Smartphones. *Information Systems Journal* 24, 4 (2014), 299–321. https://doi.org/10.1111/isj.12031

[33] Sanjay Kairam and Jeffrey Heer. 2016. Parting Crowds: Characterizing Divergent Interpretations in Crowdsourced Annotation Tasks. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work, CSCW*, Vol. 27. Association for Computing Machinery, New York, NY, USA, 1637–1648. https://doi.org/10.1145/2818048.2820016

[34] Merlin Knaeble, Mario Nadj, and Alexander Maedche. 2020. Oracle or Teacher? A Systematic Overview of Research on Interactive Labeling for Machine Learning. In *WI2020 Zentrale Tracks*, Norbert Gronau, Moreen Heine, K Poustcchi, and H Krasnova (Eds.). GITO Verlag, Berlin, Germany, 2–16. https://doi.org/10.30844/wi{_}2020{_}a1-knaeble

[35] J. Richard Landis and Gary G. Koch. 1977. The Measurement of Observer Agreement for Categorical Data. *Biometrics* 33, 1 (3 1977), 159. https://doi.org/10.2307/2529310

[36] Laura Lascau, Sandy J.J. Gould, Anna L. Cox, Elizaveta Karmannaya, and Duncan P. Brumby. 2019. Monotasking or Multitasking: Designing for Crowdworkers' Preferences. In *Conference on Human Factors in Computing Systems - Proceedings*, Stephen Brewster, Geraldine Fitzpatrick, Anna Cox, Vassilis Kostakos, and Anna L Cox (Eds.). Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3290605.3300649

[37] Walter S Lasecki, Mitchell Gordon, Danai Koutra, Malte F Jung, Steven P Dow, and Jeffrey P Bigham. 2014. Glance: Rapidly Coding Behavioral Video with the Crowd. In *UIST 2014 - Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*. Association for Computing Machinery, New York, NY, USA, 551–562. https://doi.org/10.1145/2642918.2647367

[38] Edith Law, Krzysztof Z Gajos, Andrea Wiggins, Mary L Gray, and Alex Williams. 2017. Crowdsourcing as a Tool for Research. In *CSCW'17*, Charlotte P Lee, Steve Poltrock, Louise Barkhuus, Marcos Borges, and Wendy Kellogg (Eds.). Association for Computing Machinery, New York, NY, USA, 1544–1561. https://doi.org/10.1145/2998181.2998197

[39] Nolan Lawson, Kevin Eustice, Mike Perkowitz, and Meliha Yetisgen-Yildiz. 2010. Annotating Large Email Datasets for Named Entity Recognition with Mechanical Turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk (CSLDAMT '10)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 71–79.

[40] John Le, Andy Edmonds, Vaughn Hester, and Lukas Biewald. 2010. Ensuring Quality in Crowdsourced Search Relevance Evaluation: The Effects of Training Question Distribution. In *SIGIR 2010 Workshop on Crowdsourcing for Search Evaluation*. Association for Computing Machinery, New York, NY, USA, 17–20.

[41] Christophe Lejeune. 2011. From Normal Business to Financial Crisis... and Back Again. An Illustration of the Benefits of Cassandre for Qualitative Analysis.

Forum: Qualitative Sozialforschung 12, 1 (2011), 19. https://orbi.uliege.be/handle/2268/83445

[42] Kevin E. Levay, Jeremy Freese, and James N. Druckman. 2016. The Demographic and Political Composition of Mechanical Turk Samples. *SAGE Open* 6, 1 (3 2016), 1–17. https://doi.org/10.1177/2158244016636433

[43] Jasy Suet Yan Liew, Nancy McCracken, Shichun Zhou, and Kevin Crowston. 2015. Optimizing Features in Active Machine Learning for Complex Qualitative Content Analysis. In *Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science*. Association for Computational Linguistics, Stroudsburg, PA, USA, 44–48. https://doi.org/10.3115/v1/w14-2513

[44] Chi-Jung Lu and Stuart W Shulman. 2008. Rigor and Flexibility in Computer-Based Qualitative Research: Introducing the Coding Analysis Toolkit. *International Journal of Multiple Research Approaches* 2, 1 (2008), 105–117. https://doi.org/10.5172/mra.455.2.1.105

[45] Roman Lukyanenko, Jeffrey Parsons, Yolanda F Wiersma, and Mahed Maddah. 2019. Expecting the Unexpected: Effects of Data Collection Design Choices on the Quality of Crowdsourced User-Generated Content. *MIS Quarterly* 43, 2 (2019), 623–647. https://doi.org/10.25300/MISQ/2019/14439

[46] Kathleen M MacQueen, Eleanor McLellan, Kelly Kay, and Bobby Milstein. 1998. Codebook Development for Team-Based Qualitative Analysis. *CAM Journal* 10, 2 (1998), 31–36. https://doi.org/10.1177/1525822X980100020301

[47] Megh Marathe and Kentaro Toyama. 2018. Semi-Automated Coding for Qualitative Research: A User-Centered Inquiry and Initial Prototypes. In *Conference on Human Factors in Computing Systems - Proceedings*, Regan Mandryk, Mark Hancock, Mark Perry, and Anna Cox (Eds.). Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3173574.3173922

[48] Matthew Marge, Satanjeev Banerjee, and Alexander I Rudnicky. 2010. Using the Amazon Mechanical Turk to Transcribe and Annotate Meeting Speech for Extractive Summarization. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk (CSLDAMT '10)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 99–107.

[49] Winter Mason and Duncan Watts. 2009. Financial incentives and the "performance of crowds. *SIGKDD Explorations* 11 (2009), 100–108. https://doi.org/10.1145/1600150.1600175

[50] Philipp Mayring. 2000. Qualitative Content Analysis. *Forum: Qualitative Sozialforschung* 1, 2 (2000), 159–176.

[51] Nora McDonald, Sarita Schoenebeck, and Andrea Forte. 2019. Reliability and Inter-rater Reliability in Qualitative Research: Norms and Guidelines for CSCW and HCI Practice. , 23 pages. https://doi.org/10.1145/3359174

[52] Andrew J McMinn, Yashar Moshfeghi, and Joemon M Jose. 2013. Building a Large-Scale Corpus for Evaluating Event Detection on Twitter. In *Proceedings of the 22nd ACM International Conference on Information &amp; Knowledge Management (CIKM '13)*. Association for Computing Machinery, New York, NY, USA, 409–418. https://doi.org/10.1145/2505515.2505695

[53] Susan Miles and Gene Rowe. 2004. The Laddering Technique. In *Doing Social Psychology Research*. The British Psychological Society and Blackwell Publishing Ltd, Oxford, UK, 305–343. https://doi.org/10.1002/9780470776278.ch13

[54] Tanushree Mitra, C. J. Hutto, and Eric Gilbert. 2015. Comparing Person- and Process-centric Strategies for Obtaining Quality Data on Amazon Mechanical Turk. In *Conference on Human Factors in Computing Systems - Proceedings*, Bo Begole, Jinwoo Kim, Kori Inkpen, and Woontack Woo (Eds.). Association for Computing Machinery, New York, NY, USA, 1345–1354. https://doi.org/10.1145/2702123.2702553

[55] Phu Nguyen, Juho Kim, and Robert C. Miller. 2013. Generating Annotations for How-to Videos Using Crowdsourcing. In *Conference on Human Factors in Computing Systems - Proceedings*, Wendy Mackay, Stephen Brewster, and Susanne Bødker (Eds.). Association for Computing Machinery, New York, NY, USA, 835–840. https://doi.org/10.1145/2468356.2468506

[56] Natalya F. Noy, Jonathan Mortensen, Paul R. Alexander, and Mark A. Musen. 2013. Mechanical Turk as an Ontology Engineer? Using Microtasks as a Component of an ontology-engineering workflow. In *Proceedings of the 5th Annual ACM Web Science Conference, WebSci'13*, Hugh Davis (Ed.). Association for Computing Machinery, New York, NY, USA, 262–271. https://doi.org/10.1145/2464464.2464482

[57] David Oleson, Alexander Sorokin, Greg Laughlin, Vaughn Hester, John Le, and Lukas Biewald. 2011. Programmatic Gold: Targeted and Scalable Quality Assurance in Crowdsourcing. In *Proceedings of the 11th AAAI Conference on Human Computation (AAAIWS'11-11)*. AAAI Press, Palo Alto, CA, USA, 43–48.

[58] Gabriel Parent and Maxine Eskenazi. 2010. Clustering Dictionary Definitions Using Amazon Mechanical Turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk (CSLDAMT '10)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 21–29.

[59] Michael Quinn Patton. 2001. *Qualitative Research & Evaluation Methods*. SAGE Publications, Thousand Oaks, California, USA. 381 pages. https://us.sagepub.com/en-us/nam/qualitative-research-evaluation-methods/book232962

[60] Justus Randolph. 2005. Free-Marginal Multirater Kappa Kfree: An Alternative to Fleiss Fixed-Marginal Multirater Kappa. In *Joensuu University Learning and Instruction Symposium*. Online Submission, Joensuu, Finland, 1–20.

[61] Lyn Richards. 2002. Qualitative computing—a methods revolution? *International Journal of Social Research Methodology* 5, 3 (7 2002), 263–276. https://doi.org/10.1080/13645570210146302

[62] Hauke Riesch and Clive Potter. 2014. Citizen science as seen by scientists: Methodological, epistemological and ethical dimensions. *Public Understanding of Science* 23, 1 (1 2014), 107–120. https://doi.org/10.1177/0963662513497324

[63] Tim Rietz and Alexander Maedche. 2019. LadderBot: A Requirements Self-Elicitation System. In *Proceedings of the IEEE International Conference on Requirements Engineering*, Daniela Damian, Anna Perini, and Seok-Won Lee (Eds.), Vol. 2019-Septe. IEEE, New York, NY, USA, 357–362. https://doi.org/10.1109/RE.2019.00045

[64] Tim Rietz and Alexander Maedche. 2021. Cody: An AI-Based System to Semi-Automate Coding for Qualitative Research. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 2021)*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3411764.3445591

[65] Jeffrey Rzeszotarski and Aniket Kittur. 2012. CrowdScape: Interactively visualizing user behavior and output. In *Proceedings of the 25th annual ACM symposium on User interface software and technology (ACM Digital Library)*, Rob Miller (Ed.). Association for Computing Machinery, New York, NY, USA, 55. https://doi.org/10.1145/2380116.2380125

[66] Daniel Schlagwein and Farhad Daneshgar. 2014. User Requirements of a Crowd Sourcing Platform for Researchers: Findings from a Series of Focus Groups. In *Proceedings - Pacific Asia Conference on Information Systems, PACIS 2014*. Association for Information Systems, USA, 1–11. https://aisel.aisnet.org/pacis2014/195

[67] Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y. Ng. 2008. Cheap and fast - But is it good? Evaluating Non-Expert Annotations for Natural Language Tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP '08)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 254–263.

[68] Carl Vondrick, Donald Patterson, and Deva Ramanan. 2013. Efficiently Scaling Up Crowdsourced Video Annotation: A Set of Best Practices for High Quality, Economical Video Labeling. *International Journal of Computer Vision* 101, 1 (2013), 184–204. https://doi.org/10.1007/s11263-012-0564-1

[69] Yair Wand and Richard Y. Wang. 1996. Anchoring data quality dimensions in ontological foundations. *Commun. ACM* 39, 11 (11 1996), 86–95. https://doi.org/10.1145/240455.240479

[70] Galen Weld, Esther Jang, Anthony Li, Aileen Zeng, Kurtis Heimerl, and Jon E Froehlich. 2019. Deep Learning for Automatically Detecting Sidewalk Accessibility Problems Using Streetscape Imagery. In *The 21st International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '19)*. Association for Computing Machinery, New York, NY, USA, 196–209. https://doi.org/10.1145/3308561.3353798

[71] Gregor Wiedemann. 2013. Opening up to Big Data: Computer-Assisted Analysis of Textual Data in Social Sciences. *Forum Qualitative Sozialforschung* 14, 2 (5 2013), 332–358. https://doi.org/10.17169/fqs-14.2.1949

[72] Shomir Wilson, Florian Schaub, Rohan Ramanath, Norman Sadeh, Fei Liu, Noah A. Smith, and Frederick Liu. 2016. Crowdsourcing Annotations for Websites' Privacy Policies: Can it really work?. In *25th International World Wide Web Conference, WWW 2016*, Jacqueline Bourdeau (Ed.). International World Wide Web Conferences Steering Committee, Geneva, 133–143. https://doi.org/10.1145/2872427.2883035

[73] Meliha Yetisgen-yildiz, Imre Solti, Fei Xia, and Scott Russell Halgrim. 2010. Preliminary Experience with Amazon's Mechanical Turk for Annotating Medical Named Entities. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*. Association for Computational Linguistics, Stroudsburg, PA, USA, 180–183.

[74] Xuefeng Zhang, Mingshuang Chen, and Guanqun Ji. 2019. Factors Influencing the Crowd Participation in Knowledge-Intensive Crowdsourcing. In *Proceedings of the 4th International Conference on Crowd Science and Engineering (ICCSE'19)*. Association for Computing Machinery, New York, NY, USA, 186–194. https://doi.org/10.1145/3371238.3371268