Mining Crowdsourcing Problems from Discussion Forums of Workers

Zahra Nouri and **Henning Wachsmuth** and **Gregor Engels**
Department of Computer Science, Paderborn University, Paderborn, Germany
znouri@mail.upb.de henningw@upb.de engels@upb.de

Abstract

Crowdsourcing is used in academia and industry to solve tasks that are easy for humans but hard for computers, in natural language processing mostly to annotate data. The quality of annotations is affected by problems in the task design, task operation, and task evaluation that workers face with requesters in crowdsourcing processes. To learn about the major problems, we provide a short but comprehensive survey based on two complementary studies: (1) a *literature review* where we collect and organize problems known from interviews with workers, and (2) an empirical *data analysis* where we use topic modeling to mine workers' complaints from a new English corpus of workers' forum discussions. While literature covers all process phases, problems in the task evaluation are prevalent, including unfair rejections, late payments, and unjustified blockings of workers. According to the data, however, poor task design in terms of malfunctioning environments, bad workload estimation, and privacy violations seems to bother the workers most. Our findings form the basis for future research on how to improve crowdsourcing processes.

1 Introduction

Crowdsourcing refers to the idea of providing collaborative services using the intelligence, skills, and creativity of a huge crowd (Howe, 2006). It has become a prevalent method in academia and industry to solve tasks that are complex for computers and yet easy for humans with low cost and high creativity. A typical use case in natural language processing (NLP) is the annotation of data. Crowdsourcing platforms create an extensive network of people and enable organizations and individuals (the so called task *requesters*) to hire other individuals (the *crowdworkers* or just *workers*) in an anonymous and distant manner. Among the most widely used platforms are Upwork, Appen, and above all Amazon Mechanical Turk (MTurk) which host annotation tasks and collect huge sets of annotated data from workers.

Conceptually, the general crowdsourcing process includes three main phases which are illustrated in Figure 1 and further detailed in Section 2: (1) In the *task design*, requesters create and post tasks on the given platform. (2) In the *task operation*, workers take on and solve tasks to then submit their results. Workers may ask for clarifications of task details, and requesters may give feedback to ensure correct results (Gupta et al., 2014). And (3) in the *task evaluation*, requesters decide whether to accept results and hence to pay workers, while workers may ask about feedback, payment, or reasons for rejections. To achieve mutual benefit in this process, workers need to understand the tasks designed by the requesters and their expected results. In practice, however, workers face diverse problems in the process and the interactions (Bederson and Quinn, 2011). This does not only affect the workers' feelings of pride and satisfaction (Boons et al., 2015), but it also negatively impacts the quality of the results, which in turn leads to a lack of trust between requesters and workers (Bederson and Quinn, 2011). Despite the importance of crowdsourcing for NLP, such problems are not yet well-explored within the community.

In the long run, we aim to create technology that supports making crowdsourcing processes in NLP and other contexts more successful. Towards this goal, a first substantial step is to understand what problems exist and which of them are most dominant problems. To this end, we combine two complementary

Proceedings of the 28th International Conference on Computational Linguistics, pages 6264–6276
Barcelona, Spain (Online), December 8-13, 2020

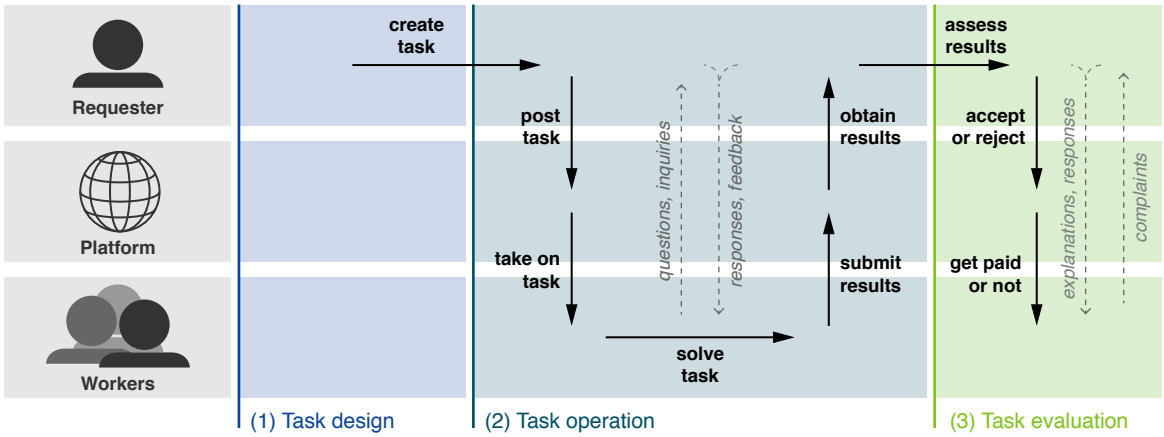

Figure 1: The three phases of a crowdsourcing process. *(1) Task design:* The requester creates a task. *(2) Task operation:* Workers accept and solve the task, and submit the results. *(3) Task evaluation:* The requester accept or reject the results. Communication may happen during task operation and evaluation.

approaches in this paper, bringing together knowledge from human computation with ideas from NLP: First, a *literature review* where we collect crowdsourcing problems discussed in related work, mostly obtained from interviews with and surveys among workers (Section 3). We consolidated the reported challenges from a wide range of studies in order to form a unified view on problems already known in literature. Second, an empirical *data analysis* where we use topic modeling to mine the workers' problems from what they complain about on a daily basis in online discussion forums (Section 4). Given its dominance, we focus on MTurk here, building a new corpus of 27,041 reviews from *Turkopticon*, a reputation portal where MTurk workers write reviews on requesters. In our analysis, we focus on the 8,610 negative reviews, since they are likely to contain complaints.[1]

The hypothesis underlying our combined approach is that the literature gives a more comprehensive view of problems existing in crowdsourcing, whereas the data better reflects their significance. In light Rof this hypothesis, we study two research questions:

RQ1. What problems do workers face in the different phases of crowdsourcing processes?

RQ2. Which of these problems are most dominant in the literature and in the data respectively?

In Section 5, we compare the results from the literature review and the data analysis. Literature covers problems in all three phases of the crowdsourcing process, along with platform-related problems. *Missing responses* and *unfair rejections* without explanation from the requesters are the most discussed problems in literature. Data reflects many problems from literature, however, the most dominant problems in the workers' complaints refer to task design: *workload misestimation*, *malfunctioning environments*, *low payment*, and *privacy violations*. Our findings imply that a thorough task design by the requesters leads to significant improvements in crowdsourcing and a better technological support may be needed.

With our research, we contribute to a better understanding of the current challenges relating to crowdsourcing processes, which forms the basis of any process improvements and best practices. Beyond the insights of this paper, the provided corpus allows for studying related questions, e.g., regarding the difference between the topics discussed in positive and in negative reviews. Also, future work may additionally consider the requesters' perspective on crowdsourcing problems. So far, however, barely any literature or data reflecting this perspective is available. Crowdsourcing is prototypical for distant work employment processes, which become increasingly common within the digitization of society. We believe that an understanding of its mechanisms is crucial to shape the work life of the future.

[1]The Turkopticon forum can be found at `https://turkopticon.info`, and the corpus can be accessed at `https://pace.uni-paderborn.de/pace-phd-programs/digital-future/resources`.

2 Crowdsourcing Processes

Crowdsourcing outsources *tasks* of a *requester* (i.e., some organization or individual) to a crowd of *workers* around the world. All communication and interaction between the requesters and workers happens via a crowdsourcing *platform*. On Amazon Mechanical Turk (MTurk), for instance, requesters may post a wide range of micro-tasks, including text annotation, translation, writing, and similar. A typical crowdsourcing process consists of the three main phases illustrated in Figure 1, often at least partly conducted in an iterative manner: (1) the *task design*, (2) the *task operation*, and (3) the *task evaluation*. We detail each phase in the following.[2]

Task Design First, the requester creates a task to be solved by the workers (e.g., the annotation of a set of English texts). Usually, the task is instantiated multiple times with different data (e.g., once for each considered text). The task design includes instructions for the workers, specifications of deadlines and paid rewards, possibly a time estimation, and similar. Besides, the requester may define qualification requirements to obtain more suitable workers (e.g., a worker may need to have a certain reputation score or to be a native English speaker). Part of the design is also the implementation of an environment (in terms of a user interface) where workers are supposed to operate the task and/or to submit their results.

Task Operation The (instances of) designed tasks are posted on the platform. Workers that meet the requirements can accept tasks and solve them, depending on their incentives, such as rewards or task types. Often, the actual work on a task is conducted directly via the task's interface; sometimes, only result submission takes place there. During task operation, workers and requesters may communicate regarding questions on the task or other inquiries by the workers. Once finished, workers may be required (e.g., on MTurk) to enter a completion code received from requesters to successfully submit their results. Requesters may provide feedback to obtained results in order to improve future result submissions.

Task Evaluation The task evaluation technically may happen as part of the task operation already, but conceptually defines the next phase: Given results submitted by a workers, a requester assesses them and can then decide to accept or reject them. Once the decisions are made, the workers get rewards for accepted results, and no rewards for rejections. Requesters may give explanations for their acceptance decisions to the workers via the platform. If a requester is not satisfied with a worker's performance for whatever reason, the requester may also block the worker, such that the latter cannot take on more tasks. Acceptance decisions and blockings affect the workers' reputation on the platform.

In cases of disagreement between requesters and workers on acceptance decisions, blocking, payment, or similar, the platform may be asked to mediate. However, as we will see below, such *platform support* does not always happen, and it entails specific problems itself.

3 Literature Review

This section presents the literature review that we conducted to collect and organize crowdsourcing problems of workers covered by existing research through interviews and surveys.

3.1 Literature

Altogether, we reviewed 102 articles in the following steps: We first searched for candidate articles with titles containing keywords such as "crowdworkers", "crowdsourcing", "crowdsourcing platforms", combined with keywords such as "difficulties", "issues", "problems", and "relationships". Based on the abstract, introduction and conclusion of the candidates, we filtered those with a focus on problems that crowdworkers face in crowdsourcing processes. We then went through their references to repeat the process for further candidates. As a result, we ended up with 33 articles that cover crowdsourcing problems either from a theoretical or empirical viewpoint, the latter being done mostly through questionnaires, interviews, or surveys on crowdsourcing platforms. In line with the data we use for the data analysis, most studies covered by the articles have been carried out on Amazon Mechanical Turk (MTurk), which is the biggest and most well-known microtask crowdsourcing platform (O'Neill and Martin, 2013).

[2]Some variations exist depending on the crowdsourcing platform. We abstract from these differences here for simplicity.

The 69 excluded publications are mainly from three different areas: (a) information systems and human computation, including studies on crowdsourcing processes, models, methodologies, the benefits and risks of their application on other domains, and quality management models for platforms and involved stakeholders, (b) psychology, including studies on behavioral and job attitudes, and (c) business and organization management, including studies on crowdsourcing models, business values, and workforce management. There are also few studies from law, and sociology.

3.2 Method

We went manually through all 33 selected articles, in order to identify the problems that are discussed theoretically or reported by the workers. For the comparison in Section 5, we also counted the occurrences of each problem. We grouped the identified problems according to the stage of the crowdsourcing process *where a problem has been caused*. This stage may be different from the stage *where a problem becomes visible*, for instance, a task design problem often becomes noticeable during task operation.

3.3 Results

As detailed in the following, we found 14 problems in the literature in total, from which 11 can be assigned to one of the three main process phases (i.e., task design, task operation, and task evaluation). The remaining three problems refer to platform support and, hence, conceptually come after the others. To refer to the problems later on, we number them in parentheses below increasingly from 1 to 14.

Task Design Problems (problems 1–5) According to the literature, workers deliver low quality results due to the (1) *ambiguous instructions*, where it is hard to understand the task and the desired result (Fowler Jr., 1992; Khanna et al., 2010; Chandler et al., 2013; Gadiraju et al., 2017; Gaikwad et al., 2017). The workers become frustrated from (2) *malfunctioning environments*, where technical errors in the task interfaces created by the requesters cause unsuccessful submissions (Silberman et al., 2010b; Bederson and Quinn, 2011; Silberman, 2015; Brawley and Pury, 2016; McInnis et al., 2016; Berg et al., 2018). They also complain about (3) *workload misestimation* where inequality of required effort and time leads to fruitless attempts to finish the tasks (Silberman et al., 2010a; Gupta et al., 2014; Silberman, 2015). This inequality consequently maybe also be reflected in (4) *low payment* (i.e., low hourly payment ratio), which affects the workers' motivation and satisfaction (Ross et al., 2010; Silberman, 2015; Gadiraju et al., 2017; Berg et al., 2018). Literature also reports on (5) *privacy violations* where tasks invade workers' privacy by collecting, processing, and misusing workers' personal information in different ways (Lease et al., 2013; Kang et al., 2014; Halder, 2014; Vakharia and Lease, 2015; Silberman, 2015; Durward et al., 2016; Edlund et al., 2017; Xia et al., 2017; Shu et al., 2018; Sannon and Cosley, 2019). Such attempts violate the term-of-service (TOS) of MTurk.

Task Operation Problems (6–8) Several studies indicate that there are often (6) *missing responses* from requesters on inquiries from workers related to tasks or desired solutions (Silberman, 2010; Silberman et al., 2010a; Silberman et al., 2010b; Bederson and Quinn, 2011; Dow et al., 2012; Chandler et al., 2013; Gupta et al., 2014; Alagarai Sampath et al., 2014; Silberman, 2015; Brawley and Pury, 2016; Deng and Joshi, 2016; Schwartz, 2018; Berg et al., 2018). Requesters often give only (7) *minor feedback* to submitted results (Dow et al., 2012; Gaikwad et al., 2017; Schwartz, 2018; Berg et al., 2018). Also, there are reports on unfriendly behavior where the requesters give (8) *mean comments* on the questions on the submitted results (Martin et al., 2014; Brawley and Pury, 2016; Xia et al., 2017; Berg et al., 2018).

Task Evaluation Problems (9–11) Many articles report on (9) *unfair rejections* of results in two ways. On one hand, requesters or automatic algorithms make (a) *harsh evaluations* of submissions rejecting results that have been created to the best of the workers' abilities (Porter, 2017; Silberman, 2010; Silberman et al., 2010a; Silberman et al., 2010b; Bederson and Quinn, 2011; Irani and Silberman, 2013; Gupta et al., 2014; Peng et al., 2014; Brawley and Pury, 2016; Guth and Brabham, 2017; Berg et al., 2018). On the other hand, there are often (b) *missing explanations* for rejections where requesters provide no or unclear reasons to the workers, making workers eventually become disappointed (Porter, 2017; Silberman et al., 2010b; Irani and Silberman, 2013; Peng et al., 2014; Gupta et al., 2014; Martin et al., 2014; Silberman,

2015; Brawley and Pury, 2016; Guth and Brabham, 2017; Berg et al., 2018). We also include (10) *late payment* here where workers wait long without being informed about the payment time (Bederson and Quinn, 2011; Silberman, 2015; Brawley and Pury, 2016). Finally, an (11) *unjustified blocking* may happen if workers ask requesters for rejection reasons or payment time (Porter, 2017; Irani and Silberman, 2013; Martin et al., 2014; Peng et al., 2014; Brawley and Pury, 2016; Guth and Brabham, 2017).

Platform Problems (12–14) Workers are allowed to report problematic cases regarding task design or the requesters' behavior to the platform. While it is neither transparent how the platform processes them, nor can it be tracked how the platform resolves the problems, workers believe that platforms perform (12) *poor mediation* in case of disagreement on rejections and payments, or complaints from workers against requesters (Silberman et al., 2010b; Khanna et al., 2010; Irani and Silberman, 2013; Vakharia and Lease, 2015; Silberman, 2015; Schwartz, 2018). In general, there is (13) *imbalanced power* between requesters and workers in that requesters are free to reject results and then not pay the workers. The workers have no support or power to change that, but have to live with the resulting lower reputation (Irani and Silberman, 2013). Moreover, platforms partly have (14) *poor tooling*. For example, requesters' contact information may not be available to workers (Chandler et al., 2013; Berg et al., 2018), and there is a lack of automated tools and proper quality control on workers' reputation, task payments, and term-of-service violations (Vakharia and Lease, 2015; Silberman, 2015). Besides, individuals are not anonymous, since workers' Amazon account is linked to their MTurk ID (Halder, 2014).

4 Data Analysis

As a complement to the literature review, we empirically analyzed reviews from a workers' online discussion forum using topic modeling, in order to investigate the most discussed problems according to the experiences that workers complain about in practice. This section summarizes our analysis as well as the corpus that we created for this purpose and for future research in the community. Our analysis aims to identify the problems that workers face in crowdsourcing processes rather than the extent to which the problems change over time. While such temporal aspects are worthy investigating given the rapid evolution of crowdsourcing, we leave them aside here for a focused discussion.

4.1 Data

On most crowdsourcing platforms, workers have no way to share their experiences made with requesters. As a result, worker community forums have come up over time, particularly for Amazon Mechanical Turk (MTurk). As such, these forums provide a suitable data set to study workers' complaints.

Forums specific to MTurk include TurkerView, MTurkCrowd, TurkerHub, TurkerNation (now on Reddit), and Turkopticon. For our analysis, we chose the latter, since Turkopticon is one of the biggest, and since it is specifically designed to support workers' rights in their relationship with requesters (Irani and Silberman, 2013). This forum functions as a reputation system where workers review requesters, reporting on experiences regarding tasks, payments, and rejections while performing a task on a daily basis. In the reviews, workers rate the requesters and either recommend or do not recommend a requester to other workers. Other workers can then read the reviews before accepting requesters' tasks.

For our corpus, we crawled 27,041 Turkopticon reviews from February 2017 to November 2018, i.e., all from the very first review on the forum until the time of crawling the data. For the present study, we selected all 8610 reviews with the tag "not recommended", since we hypothesized them to focus on complaints, particularly given the rather short average length of a review (57.2 tokens). In total, the resulting dataset comprises 492,713 tokens (15,921 unique words). The full corpus can be accessed here: `https://pace.uni-paderborn.de/pace-phd-programs/digital-future/resources`.

4.2 Method

To mine crowdsourcing problems from the reviews computationally, we relied on standard topic modeling using Latent Dirichlet Allocation (LDA). LDA finds the hidden, possibly overlapping k topics in a set of documents (Blei et al., 2003). By means of LDA, each topic is seen as a cluster of similar documents (here: reviews), modeled as a list of representative words.

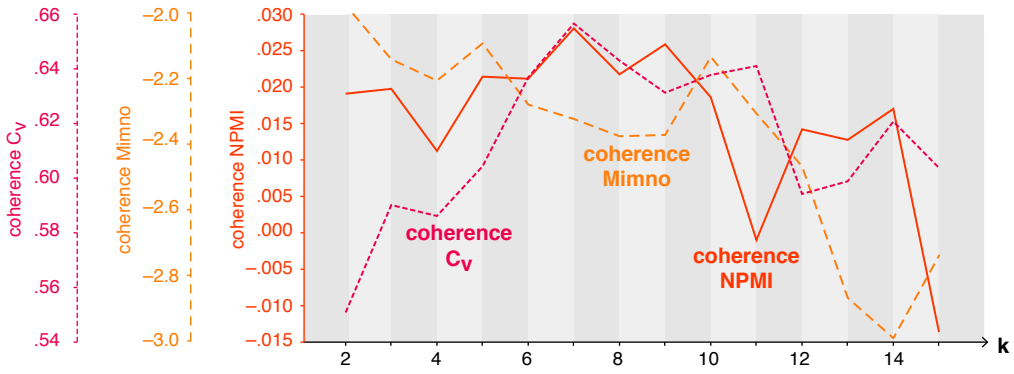

Figure 2: Topic coherence degree of our topic modeling approach according to the three applied evaluation metrics (Mimno, NPMI, and C_v) for different number of topics $k \in \{2, \ldots, 15\}$. Despite their different scales, the interpolated curves are shown in one plot, to make it easy to see the best k.

Before applying LDA, we preprocessed each review in seven subsequent cleansing steps: (1) *Tokenization* to segment each review into words and other tokens, (2) *removal of numbers* such as dates, times, and prices, (3) *removal of stopwords* such as function words (e.g., "at", "to", "which"), (4) *removal of punctuation* which we observed overproportially in the reviews, (5) *lemmatization* to merge different inflections of the same word, (6) *removal of the tag* "not-recommended" found in all negative reviews, and (7) *removal of low-frequency words*, namely those that occur only once in all 8,610 analyzed reviews. After preprocessing, the corpus size was reduced to 179,539 tokens (3808 unique words). An input instance to LDA finally spanned 20.9 tokens on average.

The performance of LDA highly depends on its parameters, including the number of topics k and the priors for the per-document topic distribution α and the per-topic word distribution η. Following Wallach et al. (2009a), we chose α asymmetrically as $1.0/k$, and η symmetrically as 0.1. This narrowed down the problem of finding the best topic model to the choice of k, which we approached empirically. We executed LDA with 500 iterations repeatedly for each $2 \leq k \leq 15$. To choose the best k among the 14 generated models, we compared the suitability of several candidate evaluation metrics.

In particular, various metrics for k exist, including *harmonic mean* (Griffiths and Steyvers, 2004), *pairwise cosine distance* (Cao et al., 2009), and *KL divergence* (Arun et al., 2010). Due to the inaccuracy of these metrics, Wallach et al. (2009b) proposed the *chib-style estimator*, which maximizes the probability of held-out documents. However, this estimator does not assess the interpretability of topics. As a solution, Chang et al. (2009) measured the topic coherence of models based on human judgments, Newman et al. (2010) developed a technique to estimate human judgments, and Mimno et al. (2011), Stevens et al. (2012), and Röder et al. (2015) studied the accuracy of several coherence metrics. Since we aim for topics that can be interpreted well in terms of the workers' problems, topic coherence metrics make most sense in the given setting. Thus, we evaluated our models based on the following metrics:

1. The *Mimno metric*, which uses top-word co-occurrence statistics and is computed in three main steps: (a) identification of distinct classes of low-quality topics, /b) identification of specific semantic problems in topic models, and (c) optimization of the topic coherence (Mimno et al., 2011).
2. *Normalized point-wise mutual information (NPMI)*, which measures the probability difference of word co-occurrences to their expectation (Bouma, 2009).
3. The *hybrid coherence measure* C_v, which is based on a combination of known approaches that is said to approximate human ratings of topic interpretability well (Röder et al., 2015).

Using the metrics, we chose k and then manually interpreted each resulting topic, as described below.

4.3 Results

Figure 2 shows that two of the three topic coherence metrics (NPMI and C_v) vote for $k = 7$, while Mimno is best for $k = 2$ and $k = 5$. We manually examined the resulting topic models for $k = \{2, 5, 7\}$

Malfunctioning environment		Workload misestimation				
1 Software errors	**2 Failed completion**	**3 Bad time estimation**	**4 High work effort**	**5 Low payment**	**6 Privacy violations**	**7 Unfair rejection**
break, link, page, one, requester, survey, return, question, amazon, problem, report, get, dead, issue, try	code, survey, break, time, submit, complete, end, timer, get, completion, minutes, waste, error, finish, take	pay, minutes, time, low, long, take, question, one, bubble, page, way, would, even, end, get	writing, write, much, cent, worth, require, penny, paragraph, back, prompt, work, photo, want, single, throw	underpay, pay, writing, screener, bad, screen, unpaid, avoid, requester, research, word, question, per, study, number	tos, require, email, violation, information, ask, site, inquisit, firefox, address, error, personal, file, name, website	reject, requester, rejection, work, work, hit, get, check, email, update, answer, response, attention, reason, one

Table 1: The top 15 words of each of the seven crowdsourcing problems found via topic modeling. The label of each problem (column headings) has been assigned manually.

and then chose $k = 7$, since it led to the model with the most meaningful and well-separated topics.

Given the seven determined topics, we first interpreted them based on the 15 highest-probability words in the associated word lists, found in Table 1. For reliability, two authors of this paper did so independently, one of which did not know the literature review results. Both then sat together to derive a problem label for each topic. To increase the certainty of a correct interpretation, we additionally inspected the 20 highest-probability reviews of each topic. Examples for each topic are shown in Table 2. This indeed led to a slight label change in one case. As for the literature review, we group the results of our data analysis according to the crowdsourcing process stages:

Task Design Problems (1–6) In the reviews, worker complain about a *malfunctioning environment*: Due to (1) *software errors* or (2) a *failed completion*, workers are not successful to submit their results and consequently do not receive any rewards. Workers also believe that tasks have a *workload misestimation* in terms of either (3) a *bad time estimation* during the task design or (4) *high work effort* required to complete the tasks. Moreover, workers complain about (5) *privacy violations* regarding the terms-of-service of MTurk where requesters ask for personal information, such as real names, social media IDs, e-mail addresses, and similar. Besides, workers report requesters offering (6) *low payment* where the tasks are simply paid very little in general or where rewards are too small for the desired solution.

Task Evaluation Problems (7) The last topic found covers (7) *unfair rejections* where workers complain that requesters are allowed to reject the results and not to pay the reward to the workers, although they still have access to the rejected results and can benefit from them. Requesters are often unresponsive to inquiries on rejections or provide unclear reasons.

As can be seen, we did not find any problem in the data analysis that explicitly refers to the *task operation* or the *platform support*. We point, though, that such problems may still appear in the reviews but are shadowed by other problems. We detail this limitation of our approach further below.

It is worth pointing out that the results of our study naturally denote only a snapshot of the problems in crowdsourcing processes for the time period covered. Our corpus contains discussed problems until November 2018. To the best of our knowledge, however, neither have the task design and communication between requesters and workers considerably changed since then, nor have effective solutions been deployed on Mturk to resolve the reported problems. Therefore, we are convinced that most insights obtained from the corpus still hold for current challenges in crowdsourcing processes.

5 Comparison of Literature Review and Data Analysis

Finally, we compare the results obtained from the data analysis to those from the literature review, in light of the two research questions from Section 1. To this end, we counted how often each problem was discussed in the literature, and we compared the relative proportions of these problems with the problem probabilities obtained from topic modeling. Figure 3 illustrates the results discussed in the following.

#	Example Reviews from Turkopticon
1	*Malfunctioning environment: Software errors*
	"Survey link goes to 404 page (hit attempted 3/30/17),I suspect there is something wrong with the URL but I couldn't find a way to edit it to get to work. Reported to Amazon and returned."
	"broken, dead link (4/28/18),The link opens to a page with a Start button. When I clicked that button I got a popup with this:,'This experiment is not currently available.',I've seen this exact page before, but I haven't posted a review about it here so it must have been for a different requester - maybe someone who is using the same broken template. In any case, the survey is not accessible so I reported it to Amazon."
2	*Malfunctioning environment: Failed completion*
	"No survey code given at the end of survey. 'We thank you for your time spent taking this survey. Your response has been recorded.' Unable to submit the HIT,notrecommended"
	"Broken hit, no response. HIT itself is fine, but there is no completion code box to putt he code once you have finished. The HIT cannot be submitted without the code, so you will be forced to return. I will update when I get a response EDIT: No response was received."
3	*Workload misestimation: Bad time estimation*
	"way too long for the pay Hit done 4/26/17,Well, the time estimate in the title was correct. Unfortunately This included a 4-minute video, a page of reading, and a whole lot of bubble questions. Too much for the pay. Stay away."
	"Way too long for the pay. Lots of video and audio to watch and click. Not hard, but just never seems to end. Bad pay."
4	*Workload misestimation: High work effort*
	"requires you to install software that tracks your internet activity and shopping, Install internet browsing tracker for $5 The HIT is to install an internet browsing tracker plus a long survey. They will track your internet browsing and shopping habits. Who does this for money? Who would let this happen?"
	"writing, not going to do it threw it back – has an open-ended writing section (at least one) of the sort I refuse to do. At least, not for pennies."
5	*Privacy violation*
	"This requester has a history of TOS violations (asking for personal info like an email address). If you decide to try this, don't give any personal info and if it asks for any, report the hit and return it."
	"Violates TOS They want turkers to register at a different website. They are collecting e-mail used to sign up. Collecting personally identifiable information. Violates TOS."
6	*Low payment*
	"very bad pay for a writing hit UNDERPAID WRITING,Writing tasks this involving must pay a lot more Avoid this cheap requester who does not fairly price writing HITs."
	"horribly underpaid -May 14th 2017-,Survey about opinions on government programs, corporate banks and regulations. Some light writing involved. Way underpaid. 2 Day AA."
7	*Unfair rejection*
	"Will reject with no possibility of resolve and not respond back I was rejected for being a male and doing the survey. I did not read anywhere that males are not allowed and he rejected my work. STAY AWAY from this requester or you will get rejected as well and will not respond to your messages."
	"Will not respond back to emails Rejected hit saying I failed to provide valid responses. I tried to contact her to get a better explanation and she has failed to respond back."

Table 2: Two example Turkopticon reviews for each crowdsourcing problem found in the data analysis.

RQ1. What problems do workers face in the different phases of crowdsourcing processes? Altogether, we found 14 different problems spanning all parts the crowdsourcing process: Five problems refer to the task design, three to the task operation, three to the task evaluation, and three to the platform support (details on the problems in Sections 3 and 4). While all problems are already covered in the literature, three of them occur in different facets in either literature or data, namely, the *malfunctioning environment*, *workload misestimation*, and *unfair rejections*. Moreover, our data analysis reveals clear differences in the perceived importance of the problems.

RQ2. Which problems are most dominant in the literature and in the data respectively? The literature covers the different phases rather similarly, with major problems being *missing responses* (14.3%) during task operation and *unfair rejections* in the task evaluation (12.1% *harsh evaluation*, 11.0% *miss-*

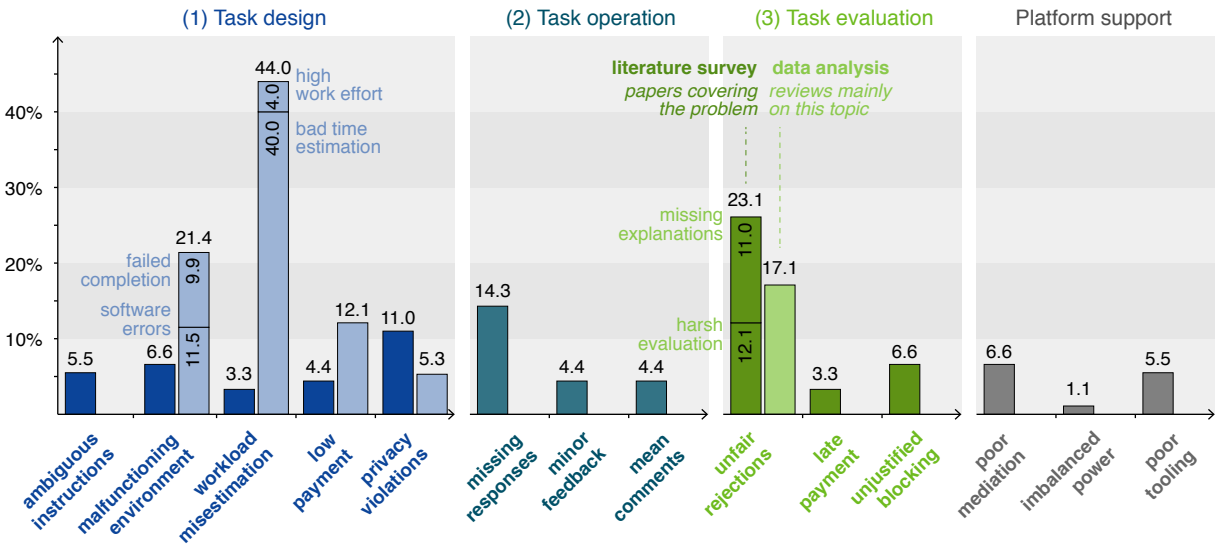

Figure 3: Distribution of the 14 found problems that workers face in the three phrases of crowdsourcing processes and with the associated platform support. *Dark bars:* Relative frequency of each problem in the reviewed literature. *Light bars:* Topic modeling probability of each problem in the data analysis.

ing explanations). Such rejections also play a big role in Turkopticon reviews (17.1%), but the major problems discussed there lie in the task design: Workers complain about *workload misestimation* (44.0% in total) and *low payment* (12.1%), along with *malfunctioning environments* (11.5% *failed completion*, 9.9% *software errors*), and *privacy violations* (5.3%) regarding the terms-of-service.

Task design and task evaluation problems are well-visible in both sources. In contrast, we did not directly observe task operation and platform support problems in the data. One reason for this lies in the limited capability of topic modeling to distinguish fine-grained and discrete topics. Consequently, our data analysis may have missed problems not necessarily because they were not discussed at all in the reviews, but because few reviews really focused on them. For example, some mention both unresponsive requesters and platform mediation problems. In contrast, discussing only the latter seems rare.

Still, since we can reasonably assume that the data reflects what most bothers the workers, our analysis reveals that the most dominant problems — from the workers' perspective — lie in the task design.

5.1 Implications

Looking at the found problems in general, we see that most of them originate in a poor task design, along with a poor communication during the task operation and evaluation. The task design is a crucial step in crowdsourcing processes that has a major effect on the quality of results (Kittur, 2010). Problems show up when requesters fail to decompose complex tasks into simple ones fairly with respect to the effort and offered reward. Poor communication negatively affect the workers' satisfaction, reputation, and achievements (Bederson and Quinn, 2011; Boons et al., 2015). Given that the workers are those who produce the results, better design and communication processes are likely to increase the quality of crowdsourcing processes as a whole. With the growing popularity of crowdsourcing, we conclude that we need such improvements to ensure a fair and effective process for both requesters and workers.

We discussed our research with a manager from MTurk who said that they recently developed some policies to overcome problems faced by requesters and workers. One recent feature resulting from this is that the platform meanwhile shows a requester's "acceptance rate" (i.e., the proportion of tasks the requester has accepted in the past) to help the workers find reliable requesters. Also, a wide range of general guidelines and policies exists on MTurk for both sides, such as task design templates, terms of service, and similar. Still, many problems remain, as our study shows.

In addition, we revisited our findings with social scientists from our research project. They suggested that requesting personal information in task design may partly originate in traditional selection mech-

anisms where employers tend to follow stereotypical views of the ideal employee to find employees according to specific requirements (Acker, 1990; Van der Lippe et al., 2019). Accordingly, requesters may circumvent the terms of service that prevent them from asking workers for personal information. The examples in Table 2 leave room for this hypothesis, but it requires further investigation in the future.

6 Conclusion

Natural language processing heavily relies on crowdsourcing, for the annotation of corpora as well as for user studies as part of approach evaluations and similar. Since the quality of crowdsourcing results is directly affected by the quality of the preceding process from task design to task evaluation, we investigate how to improve this process through computational methods. In this paper, we have combined a literature review with a data analysis, in order to identify the problems that workers face with requesters in the design, operation, and evaluation of crowdsourcing tasks. The literature covers findings from theory along with results of interviews and questionnaires that cover the workers' problems. In the data analysis, we have used topic modeling to mine respective problems from the workers' complaints in an online discussion forum, and we provide the underlying corpus for further research in the community. Bringing findings from literature and data together, we aim to bridge and share knowledge between the two research fields of human computation and natural language processing.

We found that the main problems workers complain about refer to the task design: Not only do requesters seem to underestimate the workload with respect to the time, effort, and fair payment needed to accomplish a task. Also, errors in their task environment implementation cause workers to waste time with unsuccessful submissions. Some requesters are also said to violate the workers' privacy, not respecting the terms-of-service of the used crowdsourcing platform. Unfair rejections without explanation are the dominant problem in the task evaluation, and the literature also extensively discusses communication problems between requesters and workers, combined with a poor platform support. A look at the requesters' perspective on the found problems would nicely complement our study, but literature and data for this purpose are missing so far.

Even though crowdsourcing is continuously evolving and certain aspects may change compared to the time we gathered our data, our findings show fundamental problems in different stages of the processes and are worth investigating more deeply. Approaches to tackle some problems exist already: for task design, specific interfaces help requesters create high-quality tasks; for fairer rejections, also a requester's acceptance rate was introduced, letting workers find trustworthy requesters; and for fairer requester behavior in general, discussion forums (such as Turkopticon) help support the workers' rights. However, workers still face many problems, so more improvements are required.

In future work, we seek to improve the requester-worker relationship through a technological support of the task design. In particular, we envision an automated assistant system that helps improve the clarity of task descriptions. We believe that this a key step in assisting the requesters, specially the beginners, to learn lessons from the previous requesters' mistakes in order to create more clear task descriptions, which consequently leads to receive high quality results from workers. In line with our findings, we believe that this will solve many problems of today's crowdsourcing processes.

References

Joan Acker. 1990. Hierarchies, jobs, bodies: A theory of gendered organizations. *Gender and Society*, 4(2):139–158.

Harini Alagarai Sampath, Rajeev Rajeshuni, and Bipin Indurkhya. 2014. Cognitively inspired task design to improve user performance on crowdsourcing platforms. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, page 3665–3674, New York, NY, USA. Association for Computing Machinery.

Rajkumar Arun, Venkatasubramaniyan Suresh, C.E. Veni Madhavan, and M.N. Narasimha Murthy. 2010. On finding the natural number of topics with Latent Dirichlet Allocation: Some observations. In *Advances in Knowledge Discovery and Data Mining*, pages 391–402, Berlin, Heidelberg. Springer Berlin Heidelberg.

Benjamin B. Bederson and Alexander J. Quinn. 2011. Web workers unite! Addressing challenges of online laborers. In *CHI '11 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '11, page 97–106, New York, NY, USA. Association for Computing Machinery.

Janine Berg, Marianne Furrer, Ellie Harmon, Uma Rani, and M. Six Silberman. 2018. Digital labour platforms and the future of work: Towards decent work in the online world. *Internation Labour Organization*.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022.

Mark Boons, Daan Stam, and Harry G. Barkema. 2015. Feelings of pride and respect as drivers of ongoing member activity on crowdsourcing platforms. *Journal of Management Studies*, 52(6):717–741.

Gerlof Bouma. 2009. Normalized (pointwise) mutual information in collocation extraction. *Proceedings of GSCL*, pages 31–40.

Alice M. Brawley and Cynthia LS. Pury. 2016. Work experiences on MTurk: Job satisfaction, turnover, and information sharing. *Computers in Human Behavior*, 54:531–546.

Juan Cao, Tian Xia, Jintao Li, Yongdong Zhang, and Sheng Tang. 2009. A density-based method for adaptive LDA model selection. *Neurocomputing*, 72(7):1775 – 1781. Advances in Machine Learning and Computational Intelligence.

Jesse Chandler, Gabriele Paolacci, and Pam Mueller, 2013. *Risks and Rewards of Crowdsourcing Marketplaces*, pages 377–392. Springer New York, New York, NY.

Jonathan Chang, Sean Gerrish, Chong Wang, Jordan L. Boyd-Graber, and David M. Blei. 2009. Reading tea leaves: How humans interpret topic models. In *Advances in Neural Information Processing Systems*, pages 288–296.

Xuefei Nancy Deng and Kshiti D. Joshi. 2016. Why individuals participate in micro-task crowdsourcing work environment: Revealing crowdworkers' perceptions. *Journal of the Association for Information Systems*, 17(10):648.

Steven Dow, Anand Kulkarni, Scott Klemmer, and Björn Hartmann. 2012. Shepherding the crowd yields better work. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work*, CSCW '12, page 1013–1022, New York, NY, USA. Association for Computing Machinery.

David Durward, Ivo Blohm, and Jan Marco Leimeister. 2016. Is there PAPA in crowd work?: A literature review on ethical dimensions in crowdsourcing. In *2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld)*, pages 823–832. IEEE.

John E. Edlund, Kathleene M. Lange, Andrea M. Sevene, Jonathan Umansky, Cassandra D. Beck, and Daniel J. Bell. 2017. Participant crosstalk: Issues when using the Mechanical Turk. *The Quantitative Methods for Psychology*, 13(3):174–182.

Floyd Jackson Fowler Jr. 1992. How unclear terms affect survey data. *Public Opinion Quarterly*, 56(2):218–231, 01.

Ujwal Gadiraju, Jie Yang, and Alessandro Bozzon. 2017. Clarity is a worthwhile quality: On the role of task clarity in microtask crowdsourcing. In *Proceedings of the 28th ACM Conference on Hypertext and Social Media*, HT '17, page 5–14, New York, NY, USA. Association for Computing Machinery.

Snehalkumar (Neil) S. Gaikwad, Mark E. Whiting, Dilrukshi Gamage, Catherine A. Mullings, Dinesh Majeti, Shirish Goyal, Aaron Gilbee, Nalin Chhibber, Adam Ginzberg, Angela Richmond-Fuller, Sekandar Matin, Vibhor Sehgal, Tejas Seshadri Sarma, Ahmed Nasser, Alipta Ballav, Jeff Regino, Sharon Zhou, Kamila Mananova, Preethi Srinivas, Karolina Ziulkoski, Dinesh Dhakal, Alexander Stolzoff, Senadhipathige S. Niranga, Mohamed Hashim Salih, Akshansh Sinha, Rajan Vaish, and Michael S. Bernstein. 2017. The Daemo crowdsourcing marketplace. In *Companion of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, CSCW '17 Companion, page 1–4, New York, NY, USA. Association for Computing Machinery.

Thomas L. Griffiths and Mark Steyvers. 2004. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(suppl 1):5228–5235.

Neha Gupta, David Martin, Benjamin V. Hanrahan, and Jacki O'Neill. 2014. Turk-life in India. In *Proceedings of the 18th International Conference on Supporting Group Work*, GROUP '14, page 1–11, New York, NY, USA. Association for Computing Machinery.

Kristen L. Guth and Daren C. Brabham. 2017. Finding the diamond in the rough: Exploring communication and platform in crowdsourcing performance. *Communication Monographs*, 84(4):510–533.

Buddhadeb Halder. 2014. Evolution of crowdsourcing: Potential data protection, privacy and security concerns under the new media age. *Revista Democracia Digital e Governo Eletrônico*, 1(10):377–393.

Jeff Howe. 2006. The rise of crowdsourcing. *Wired magazine*, 14(6):1–4.

Lilly C. Irani and M. Six Silberman. 2013. Turkopticon: Interrupting worker invisibility in Amazon Mechanical Turk. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, page 611–620, New York, NY, USA. Association for Computing Machinery.

Ruogu Kang, Stephanie Brown, Laura Dabbish, and Sara Kiesler. 2014. Privacy attitudes of Mechanical Turk workers and the U.S. public. In *10th Symposium On Usable Privacy and Security (SOUPS 2014)*, pages 37–49, Menlo Park, CA, July. USENIX Association.

Shashank Khanna, Aishwarya Ratan, James Davis, and William Thies. 2010. Evaluating and improving the usability of Mechanical Turk for low-income workers in India. In *Proceedings of the First ACM Symposium on Computing for Development*, ACM DEV '10, New York, NY, USA. Association for Computing Machinery.

Aniket Kittur. 2010. Crowdsourcing, collaboration and creativity. *XRDS: crossroads, the ACM Magazine for Students*, 17(2):22–26.

Matthew Lease, Jessica Hullman, Jeffrey Bigham, Michael Bernstein, Juho Kim, Walter Lasecki, Saeideh Bakhshi, Tanushree Mitra, and Robert Miller. 2013. Mechanical Turk is not anonymous. *Available at SSRN 2228728*.

David Martin, Benjamin V. Hanrahan, Jacki O'Neill, and Neha Gupta. 2014. Being a turker. In *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work and Social Computing*, CSCW '14, pages 224–235, New York, NY, USA. Association for Computing Machinery.

Brian McInnis, Dan Cosley, Chaebong Nam, and Gilly Leshed. 2016. Taking a HIT: Designing around rejection, mistrust, risk, and workers' experiences in Amazon Mechanical Turk. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, page 2271–2282, New York, NY, USA. Association for Computing Machinery.

David Mimno, Hanna M. Wallach, Edmund Talley, Miriam Leenders, and Andrew McCallum. 2011. Optimizing semantic coherence in topic models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, page 262–272, USA. Association for Computational Linguistics.

David Newman, Jey Han Lau, Karl Grieser, and Timothy Baldwin. 2010. Automatic evaluation of topic coherence. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, page 100–108, USA. Association for Computational Linguistics.

Jacki O'Neill and David Martin. 2013. Relationship-based business process crowdsourcing? In Paula Kotzé, Gary Marsden, Gitte Lindgaard, Janet Wesson, and Marco Winckler, editors, *Human-Computer Interaction – INTERACT 2013*, pages 429–446, Berlin, Heidelberg. Springer Berlin Heidelberg.

Xin Peng, Muhammad Ali Babar, and Christof Ebert. 2014. Collaborative software development platforms for crowdsourcing. *IEEE Software*, 31(2):30–36.

Constance Elise Porter. 2017. A typology of virtual communities: A multi-disciplinary foundation for future research. *Journal of Computer-Mediated Communication*, 10(1), 07. JCMC1011.

Michael Röder, Andreas Both, and Alexander Hinneburg. 2015. Exploring the space of topic coherence measures. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, WSDM '15, page 399–408, New York, NY, USA. Association for Computing Machinery.

Joel Ross, Lilly Irani, M. Six Silberman, Andrew Zaldivar, and Bill Tomlinson. 2010. Who are the crowdworkers? Shifting demographics in Mechanical Turk. In *CHI '10 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '10, page 2863–2872, New York, NY, USA. Association for Computing Machinery.

Shruti Sannon and Dan Cosley. 2019. Privacy, power, and invisible labor on Amazon Mechanical Turk. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, page 1–12, New York, NY, USA. Association for Computing Machinery.

David Schwartz. 2018. Embedded in the crowd: Creative freelancers, crowdsourced work, and occupational community. *Work and Occupations*, 45(3):247–282.

Jiangang Shu, Xiaohua Jia, Kan Yang, and Hua Wang. 2018. Privacy-preserving task recommendation services for crowdsourcing. *IEEE Transactions on Services Computing*.

M. Six Silberman, Lilly Irani, and Joel Ross. 2010a. Ethics and tactics of professional crowdwork. *XRDS: Crossroads, The ACM Magazine for Students*, 17(2):39–43, December.

M. Six Silberman, Joel Ross, Lilly Irani, and Bill Tomlinson. 2010b. Sellers' problems in human computation markets. In *Proceedings of the ACM SIGKDD Workshop on Human Computation*, HCOMP '10, page 18–21, New York, NY, USA. Association for Computing Machinery.

M. Six Silberman. 2010. What's fair? Rational action and its residuals in an electronic market. *Unpublished manuscript*.

M. Six Silberman. 2015. *Human-centered computing and the future of work: Lessons from Mechanical Turk and Turkopticon, 2008-2015*. Ph.D. thesis, UC Irvine.

Keith Stevens, Philip Kegelmeyer, David Andrzejewski, and David Buttler. 2012. Exploring topic coherence over many models and many topics. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, page 952–961, USA. Association for Computational Linguistics.

Donna Vakharia and Matthew Lease. 2015. Beyond Mechanical Turk: An analysis of paid crowd work platforms. *Proceedings of the iConference*, pages 1–17.

Tanja Van der Lippe, Leonie Van Breeschoten, and Margriet Van Hek. 2019. Organizational work–life policies and the gender wage gap in european workplaces. *Work and Occupations*, 46(2):111–148.

Hanna M. Wallach, David M. Mimno, and Andrew McCallum. 2009a. Rethinking LDA: Why priors matter. In Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 1973–1981. Curran Associates, Inc.

Hanna M. Wallach, Iain Murray, Ruslan Salakhutdinov, and David Mimno. 2009b. Evaluation methods for topic models. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 1105–1112, New York, NY, USA. Association for Computing Machinery.

Huichuan Xia, Yang Wang, Yun Huang, and Anuj Shah. 2017. "Our privacy needs to be protected at all costs": Crowd workers' privacy experiences on Amazon Mechanical Turk. *Proceedings of ACM Human Computer Interaction*, 1(CSCW), December.