Middle East Technical University
Informatics Institute

COST MODELS OF CROWDSOURCING QUALITY ASSURANCE MECHANISMS

Advisor: Prof. Dr. Semih Bilgen
(METU)

Deniz İren
(IS-IS)

December 2013

TECHNICAL REPORT
METU/II-TR-2013-

Orta Doğu Teknik Üniversitesi
Enformatik Enstitüsü

KİTLE KAYNAKLI ÇALIŞMA KALİTE GÜVENCE YÖNTEMLERİ İÇİN MALİYET MODELLERİ

Danışman: Prof. Dr. Semih Bilgen
(ODTÜ)

Deniz İren
(IS-IS)

Aralık 2013

TEKNİK RAPOR
ODTÜ/EE-TR-2013-21

REPORT DOCUMENTATION PAGE

1. AGENCY USE ONLY (Internal Use)	**2. REPORT DATE** 31.12.2013
3. TITLE AND SUBTITLE COST MODELS OF CROWDSOURCING QUALITY ASSURANCE MECHANISMS	
4. AUTHOR (S) Deniz İren	**5. REPORT NUMBER (Internal Use)** **METU/II-TR-2013-**
6. SPONSORING/ MONITORING AGENCY NAME(S) AND SIGNATURE(S) Information Systems Programme, Department of Information Systems, Informatics Institute, METU Advisor: Semih Bilgen Signature:	
7. SUPPLEMENTARY NOTES	

8. ABSTRACT (MAXIMUM 200 WORDS)

Crowdsourcing is a business model which allows practitioners to access a rather cheap and scalable workforce. However, due to loose worker-employer relationships, skill diversity of the crowd and the anonymity of the participants, it tends to result in lower quality compared to traditional way of doing work. Thus crowdsourcing practitioners use certain mechanisms to make sure the end product complies with the quality requirements. Each quality assurance mechanism used in crowdsourcing impacts the project cost and schedule. Crowdsourcing practitioners need well defined ways to estimate these impacts in order to manage the crowdsourcing process effectively and efficiently. This technical report presents the cost models of major quality assurance techniques that may be applied in crowdsourcing and describes the cost of quality approach for analyzing the quality related costs in crowdsourcing.

9. SUBJECT TERMS	**10. NUMBER OF PAGES**
Crowdsourcing, Cost of Quality, Project Management	**39**

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Cost Models of Crowdsourcing Quality Assurance Mechanisms

Deniz İren, Semih Bilgen

Middle East Technical University

Informatics Institute

diren@metu.edu.tr | bilgen@metu.edu.tr

Abstract

Crowdsourcing is a business model which allows practitioners to access a rather cheap and scalable workforce. However, due to loose worker-employer relationships, skill diversity of the crowd and the anonymity of the participants, it tends to result in lower quality compared to traditional way of doing work. Thus crowdsourcing practitioners use certain mechanisms to make sure the end product complies with the quality requirements. Each quality assurance mechanism used in crowdsourcing impacts the project cost and schedule. Crowdsourcing practitioners need well defined ways to estimate these impacts in order to manage the crowdsourcing process effectively and efficiently. This technical report presents the cost models of major quality assurance techniques that may be applied in crowdsourcing and describes the cost of quality approach for analyzing the quality related costs in crowdsourcing.

Keywords: Crowdsourcing, Cost of Quality, Project Management;

1. INTRODUCTION

Crowdsourcing is a process of value creation as a result of an outsourcing initiative, in which the interactive features of the Internet are utilized, by generally an anonymous mass consisting of individuals voluntarily choosing the task to work on. The agreement between the employer and the workers either does not exist or is not strictly binding. The lack of tight employee-employer relationship, the anonymity and the diverse skill set of the crowd cause a tendency in crowdsourcing process to result with poor quality products. Thus, crowdsourcing researchers and practitioners use mechanisms to ensure that the end products satisfy the quality requirements. Introducing and maintaining quality assurance mechanisms result in increased project costs.

The organization of this technical report is as follows: Section 2 provides an overview of quality assurance mechanisms and Section 3 describes the cost of quality approach. Section 4 presents the cost models along with the corresponding quality assurance mechanisms. Finally, Section 5 discusses how the models can be used in real life scenarios and how these models may be used to deal with the inefficiencies of crowdsourcing in the future.

2. QUALITY ASSURANCE MECHANISMS

The crowdsourcing literature includes descriptions of many techniques to assure quality. Depending on the type of the work, these techniques offer varying range of effectiveness and applicability. Researchers frequently use different names, based on the business domain, referring to basically the same quality assurance mechanisms. In this technical report we group these mechanisms according to their characteristics which have effect on project costs. Table 1 categorizes the quality assurance mechanisms under respective groups.

Table 1: Quality assurance mechanisms and respective groups

Redundancy	**Control Group**	**Gold Standard**	**Worker Characteristics**	**Design Characteristics**
Agreement	Control group	Gold standard	Reputation	Defensive task design
Majority voting	Verification	Ground truth seeding	Expertise	Economic models
Majority decision	Validation review	Injection	Selective assignment	Statistical filtering
Multiple annotations	Community based peer reviews			Automatic check
Repeated labeling	Majority vote with comparing review			Bias / error distinction and recovery
Redundancy	Improving review (IR)			
	Majority vote with improving review (MVIR)			
	Grading			
	Multilevel review			
	Iterative improvements			
	Voting			

Following part of this section presents descriptions of the quality assurance mechanisms in groups.

2.1 Redundancy

The quality assurance mechanisms which involve assigning the same task to contributors in order to produce interchangeable results are classified as redundancy. Generally redundancy quality assurance process consists of steps shown in Figure 1.

Figure 1: Redundancy quality assurance mechanisms.

Work is broken down into multiple microtasks to be distributed to workers. Multiple instances of the same microtask are assigned to multiple workers. Workers perform the

tasks separately. Multiple results are aggregated and brought together to build the final product.

The aggregation step consists of selection of the result with best perceived quality among the same set of submissions produced as a result of completing the instances of the same microtask. Selection can be made automatically or by human cognition. Automatic selection is possible when the tasks are of *deterministic* nature, which means the same result is produced each time when the task is completed perfectly complying to the task definition (Kern, 2010). The task needs to be *objective* in order to be deterministic. In some cases even if the task is *subjective* automatic aggregation is possible. In those cases the possible outcomes of the tasks must constitute a finite set.

For example, counting the number of road junctions on a satellite image of a town is an objective determinist task. Each worker assigned with the same instance of this task counts the exact same number, if s/he does the job successfully and in good faith. On the other hand, evaluating whether a hand drawn picture resembles the figure of a cat or not and submitting a vote for or against it, is a subjective task which has a finite set of potential results. Even if the workers performing the same instance of this task are looking at the same image, they may reach different conclusions. The potential outcome of this task is binary, either positive or negative. Thus, the frequency of the votes casted for the same task instance can be calculated and the result can be automatically aggregated by selecting the majority vote. Reading a page of text and summarizing it with a couple of sentences is another example of subjective tasks, yet without a finite set of results. In this case the results can only be aggregated manually. Table 2 displays a few more examples for task types.

Table 2: Examples of task types

Objective, deterministic tasks	**Subjective tasks with finite potential result set**	**Subjective tasks with infinite (or very large) potential result set**
Transcribing an image of a distorted text	Judge an image's relevance to a text and map	Annotating a data object
Grouping similar items in a set of items	Answering a demographics survey	Tagging an image
Extract purchased items from a shopping receipt	Rating the traffic jam on a video stream of a road	Drawing an illustration of a cat
Finding duplicate items in a list	Choosing the best picture among a few pictures	Recommending a book or a movie related with given tags
Audio transcription of a news clip	Rating a product	Providing textual review about a product

When aggregation is done manually it takes the form of an evaluation. When a different party controls the quality of work outputs this mechanism is classified as a Control Group quality assurance method. Control Group quality assurance mechanisms can be used in aggregation step of Redundancy quality assurance methods.

Regardless of the aggregation method, if the quality assurance mechanism leads to assigning multiple instances of the same task, it is a redundancy quality assurance mechanism. The names given to redundancy quality assurance mechanisms by researchers vary according to the aggregation mechanism used or the business domain.

Researching on knowledge discovery, Sorokin et al. studied image annotations. They collected **multiple annotations** for the same image and used a consistency score to select the best annotations (Sorokin, 2008). Similarly, Sheng et al. collected multiple labels for data items by

repeated labeling (Sheng, 2008). The terms **majority voting** (Eagle, 2009) and **majority decision** (Hirth, 2011) are the most common names given to express redundancy.

At this point it is imperative to distinguish the main task and the quality control tasks to overcome a potential ambiguity. The main task is the task which is done in order to produce the ultimate goal of the work whereas controlling task is done to check if the results of the main task fit certain requirements or not. Both the main task and the controlling task can be assigned redundantly. In order to form a common terminology it is advised to use the term voting in a way which aligns to its real meaning, which is discussed in detail in subsection 2.2 of this technical report. This distinction is also important when choosing the correct cost model. Bottom line is control group quality assurance mechanisms can be used in aggregation steps of redundancy quality assurance mechanisms.

Submissions to different instances of the same microtask can either be made asynchronously or synchronously. Redundancy quality assurance mechanisms which seek agreement of multiple contributors are considered synchronous**. Output agreement** quality assurance mechanism which is used in ESP Game (vonAhn, 2004) requires two players to submit the same labels for the same images synchronously while **input agreement** mechanism used in Tag-a-Tune (vonAhn, 2008) required the contributors to agree upon a contribution.

Some other redundancy quality assurance mechanisms use statistical methods in aggregation step. **Inter-annotator agreement** uses Pearson correlation of individual submissions in selecting the most suitable result among submissions made when completing different instances of the same task (Snow, 2008), (Paiement, 2010).

All redundancy quality assurance mechanisms have the common characteristic that a number of instances of the same task are assigned to multiple contributors. The number of redundant submissions varies due to many reasons such as the quality requirements, cost considerations, crowd characteristics or domain constraints. Sheng et al. show that increasing the number of redundancy is only beneficial if the probability of *correctness of individual submissions* (*p*) is greater than 0.5. The level of benefit of adding more contributors changes according to this *p* value (Sheng, 2008). Zhai et al uses an iterative approach to assign weights to user votes when

deciding on crowd consensus. Certain workers have more influence on the consensus based on their former accuracy (Zhai, 2012). Using a weighted voting scheme may result in decreasing the votes needed therefore decreasing the costs.

As redundancy, by design, can lead to lowering resource efficiency to a great extent, usage of cost models for quality assurance mechanisms when designing crowdsourcing tasks is a vital way of optimizing resource utilization.

2.2 Control group

The techniques by which the submissions of the main group of workers are controlled by a separate group are control group quality assurance techniques (Figure 2). The simplest forms of controlling are voting and rating. Voting is the act of indicating a choice among a set of similar options. In crowdsourcing voting refers to a separate task that is carried out by a different group of people than the ones performing the main task. Generally voting is done at a binary nominal scale. (Yes/No, Pass/Fail, Like/NA, Selected/Unselected) As a result of the voting process, the items (tasks, products, etc.) which had the vote of the crowd can be considered of acceptable quality. Rating is defined as *classification or ranking something based on a comparative assessment* (Oxford Dictionary). Rating can be done at ordinal scale where the notion of ordering is meaningful.

When the controlling party consists of more than one individual, the controlling group needs to reach a consensus. By design, these cases pose redundancy, and the same mechanisms of aggregation apply in control tasks.

Figure 2: Control group quality assurance mechanisms

Generally controlling the outputs of a task is far less complex than performing that task. In those cases, control tasks may cost significantly less than the main task (Kern, 2010). However when the primary task is extremely simple and small, time and cost spent on verifying the task outputs become comparable with the resources used for the primary task (Ipeirotis, 2010). Hirth et al. show that using Control Group mechanisms is more cost effective when the primary task is significantly more complex than the control task (Hirth, 2013).

The control group may not only be responsible for approving and denying the submissions but also providing feedback, rationale for the decision made or improving the submission (Kern, 2010). Obviously, these additional efforts result in increased task complexity and costs.

Voting and rating can be applied in reviewing the outputs of both simple and complex tasks. Kittur et. al exemplify the usage of voting mechanism as a way of evaluating the quality of multiple Wikipedia articles which are similar in content (Kittur, 2011). Most of the contest-type crowdsourcing initiatives use voting and rating to select the best submissions. For instance, Threadless, a popular crowdsourcing initiative which focuses on t-shirt design, uses rating mechanism for selecting designs to be produced (Threadless). Rating is used in almost all online marketplaces which utilize a recommender system.

2.3 Gold standard

Also referred to as ground truth seeding (Quinn, 2011), gold standard is basically a set of trusted inputs (labels, annotations, etc.) inserted among the data, which constitute expected results for certain tasks. If contributions of a worker deviate significantly from the trusted, -gold standard- result, measures are taken to improve quality (Sorokin, 2008), (McCann, 2008), (Huang, 2010). The worker can be provided with immediate feedback including the gold standard answer to ensure that expectations are understood clearly by the worker (Ipeirotis, 2010). This has an improving effect on worker submission quality, whether the gold standard comparison is made for training the user before moving on to the real tasks (Le, 2010), or randomly carried out within the task performing process. Incompatible submissions of workers are tracked to reveal a potential pattern in order to identify cheaters. Submission patterns of workers are used to define individual reputation which can be used to establish a trust evaluation infrastructure for the crowdsourcing system or platform (Voyer, 2010).

Figure 3: Gold standard quality assurance mechanisms

Checking gold standard verification can be done at different points in crowdsourcing process. Most frequent usage is *asynchronous*, in which gold standard tasks are assigned to workers randomly in the task sequence. McCann et al. defines a mechanism for identifying trusted and

untrusted workers by using gold-standard questions (McCann, 2008). In *synchronous* usage, the main task and the gold standard task are assigned at the same time (Figure 3). As an example, ReCaptcha provides the user with images of two words together. One of the images displays a *control word* which is known in advance. If that word is submitted correctly by the user, only then the submission for the *unknown word* is considered valid. The second word is the one which is expected to be digitized (vonAhn, 2008). Gold standard tasks can be assigned to the worker before the main tasks as a method for training or evaluating competency of the worker.

The sample size of gold standard tasks must be large enough, so that probability of the same worker to be assigned with the same gold standard tasks within process is quite low. However, establishing a large gold standard data set can result in significant increases in cost. In some cases the gold standard task pool can be enriched by dynamically altering the pool content (vonAhn, 2008), (Oleson, 2011).

2.4 Worker centric

Since most of the low quality work comes from a small percentage of workers (Quinn, 2011), (Kittur, 2008a), by identifying and removing this small portion from the system, overall quality can be increased.

Quality assurance mechanisms based on worker characteristics can only be used in cases which the workers do not have total anonymity. Researchers focused their attention on a large spectrum of areas to develop ways to improve crowdsourcing quality by studying workers. These areas include but not limited to crowd demographics (Ipeirotis, 2010), (Ross, 2010), participation inequality (Stewart, 2010), contributor biases (Antin, 2012), worker character stereotypes (Kazai, 2011) and motivation (Rogstadius, 2011), (Shaw, 2011).

Reputation which is a measure of worker trustworthiness is calculated by former submissions made by the individual. Reputation can be used as criteria for selecting crowd members or identifying and banning cheaters from the crowd. Establishing reputation tracking infrastructure requires the workers to be identified by the system. Crowdsourcing platforms such as

Microworkers and Amazon Mechanical Turk keep worker accounts. Wikipedia uses a reputation system to choose reputable workers for reviewers and editors (Stvilia, 2008).

AddHoc, temporary reputation systems may also be developed. Callison-Burch used a few initial -gold standard- questions to judge if the workers are trustworthy or not. The workers were assigned trust scores according to the extent to which their answers match the expert answers (Callison-Burch, 2009).

Recently an increasing number of researchers started working on quality assurance mechanisms based on worker characteristics. These novel methods aim at managing the worker skills, biases and trustworthiness to select the most appropriate workers for specific tasks (Ho, 2012), (Difallah, 2013). Furthermore, workers' social media profiles and networks are used for worker recommendation.

2.5 Design centric

Quality assurance can be achieved through designing user friendly and more robust tasks. **Defensive task design** in crowdsourcing suggests designing the tasks in a way that cheating is not easier than completing the task in good faith (Quinn, 2011), (Kittur, 2008a). It is also recommended to include verifiable parts in tasks (Kittur, 2008a), so that statistical quality control of a task sample becomes possible.

Through a **statistical approach**, Ipeirotis emphasizes the distinction between a predictable error (or bias) and unrecoverable error (spam submission). Based on an algorithm using a confusion matrix and soft labeling technique they are able to calculate the error rate and expected cost of a contribution of a particular worker. Identifying bias patterns make recovery possible, thus decreases costs of making non-true contributions (Iperiotis, 2010).

In certain situations where a time consuming task such as reading a long text or making a hand drawing of an item, monitoring the time-to-complete the task may be valid way to detect cheaters. In an experiment involving reading and grading Wikipedia articles, Kittur et al. used time-to-complete measurements to differentiate participants who are cheating, during post

analysis of the submissions (Kittur, 2008a). When used real-time within crowdsourcing process, monitoring time-to-complete can be an effective way to identify cheaters (Xia, 2012). Tasks can be designed to last no less than a certain amount of time, and the submissions which were made faster can either be flagged for further quality control or denied automatically.

The task size can make a difference in the quality of worker contributions (Hossfeld, 2011). Thus, optimal granularity level should be achieved by dividing complex tasks into smaller, simpler, shorter microtasks.

Aside from these, crowdsourcing practitioners developed good practices and guidelines for effective task design. The task should contain **clear instructions**. The user interface should be simple and user friendly. In paid microtask crowdsourcing projects the payment must be fair. Workers tend to choose to work on tasks which they are able to perform multiple times, to maximize their gains. Enabling workers to complete tasks over and over again can result in faster task completion but also can attract cheaters.

Since one of the key characteristics of crowdsourcing is that workers freely choose the task they wish to perform, crowdsourcing practitioners should advertise the task well. Advertising a task can be simply assigning it a good, representative and interesting title and annotating the task with suitable tags in a crowdsourcing platform.

A submission which is aligned with the majority decision may not always be of high quality. Thus, denying payment for tasks which do not reflect majority decision should result in changes in crowd behavior. Being aware of that payment scheme, participants may choose to make contributions which they think that aligns with other people's submissions, not what they thing is correct.

2.6 Using multiple quality assurance mechanisms together

Using multiple quality assurance mechanisms together is a common practice, especially when desired quality level is high. However this may result in significant increases in cost. Thus

collective usage of quality assurance mechanisms should be optimized according to quality needs.

McCann et al. describes a series of quality assurance practices used together in an experiment. Acknowledging the fact that untrustworthy contributors exist in the crowd, first they try to select the trusted users by asking them evaluation questions. If the user provides enough valid answers, they classify the user as *trusted*. They collect the answers submitted to other questions which are asked to multiple users. They select the answer which was submitted most frequently by the contributors (McCann et al., 2008). In this example, using of the evaluation questions which have answers known in advance is basically a gold standard quality assurance mechanism. Establishing reputation scores for workers is *worker characteristics* type of quality assurance. Asking multiple instances of the same question to multiple people is *redundancy*. Selecting the best result among all answers is *majority decision aggregation*.

3. CROWDSOURCING COST OF QUALITY

Cost of crowdsourcing largely depends on the design decisions made during the development of the crowdsourcing system. These decisions include quality assurance mechanisms among many other parameters such as good practices, the characteristics of the selected crowd, incentives, type of work, type of crowdsourcing and selection of crowdsourcing platform. To render crowdsourcing more manageable by making the cost, time and quality estimable, these design decisions should be studied thoroughly and their effects on cost, time and quality of the project should be estimated.

It should be noted that even if the work involves no monetary payment, and a crowd is performing tasks for some other reason, workforce remains a scarce resource. Deciding to spend effort for quality assurance purposes rather than performing new tasks introduces an opportunity cost. Especially in enterprise crowdsourcing (Vukovic, 2009), significant hidden costs exist, since the crowd consists of an organization's personnel whose primary job is not performing the crowdsourced tasks, and effort not spent on primary jobs results in lost revenue for the organization.

Furthermore, when the quality assurance process and production process are separated, coordination of these processes becomes an issue. Kittur et al. examine the coordination dependencies of quality assurance methods used in crowdsourcing complex tasks. Process involves dividing the complex tasks into simple microtasks (map process), assigning them to the crowd, selecting valid outputs (reduce process) and aggregate the outputs to produce the final product (Kittur, 2011).

This section introduces Cost of Quality as an approach to analyze crowdsourcing quality assurance costs. Cost of Quality is defined as the overall cost undertaken for assuring the quality of a work product. It is expressed as the sum of *conformance* and *non-conformance* costs. Conformance costs refer to costs associated with the prevention of poor quality, whereas non-conformance costs are the costs incurred due to poor quality (Crosby, 1979). Quality appraisal and defect prevention costs are considered as conformance costs whereas costs incurred due to

errors surfaced after product delivery, non-detected errors yet to be found, non-conformances detected via quality assurance measures and rework performed to fix detected non-conformances as non-conformance costs. Due to difficulties of governing a crowd of workers, the percentage of the cost of quality in an overall crowdsourcing job is generally higher compared to the traditional way of doing business. Major CoQ categories and example crowdsourcing scenarios are listed in Table 3.

Table 3: Major types of cost of quality and examples in a crowdsourcing setting

Type	Description	Example in a crowdsourcing setting
Cost of Conformance		
- Prevention costs	Costs incurred in activities to prevent the end result from failing the quality requirements	Robust design, fitting granularity, easy to use interface
- Appraisal costs	Costs incurred to finding errors	Using a control group to detect faulty submissions
Cost of Non-conformance		
- Internal Failure (rework + retest)	Costs incurred due to non-conformances detected via quality assurance measures	Reassigning a microtask instance because the worker fails to make a submission which complies with the gold standard
- External Failure (errors emerge)	Errors surfaced after product delivery	Majority of the people translating the same work makes a deliberate cheat attempt and the wrong translation is displayed on a user's screen
- External Failure (other)	Harm done to the community or trust mechanisms	Attracting cheaters by continuously failing to detect cheat attempts, or discouraging honest contributors by frequently denying high quality submissions by mistake

Aim of any attempt on improvement of quality is not limited with achieving quality but doing it at the lowest possible cost (Schiffauerova, 2006). Numerous studies in the literature address cost optimization of common quality assurance mechanisms (Karger, 2011), (Hirth, 2013), (Okubo, 2013), (Welinder, 2010).

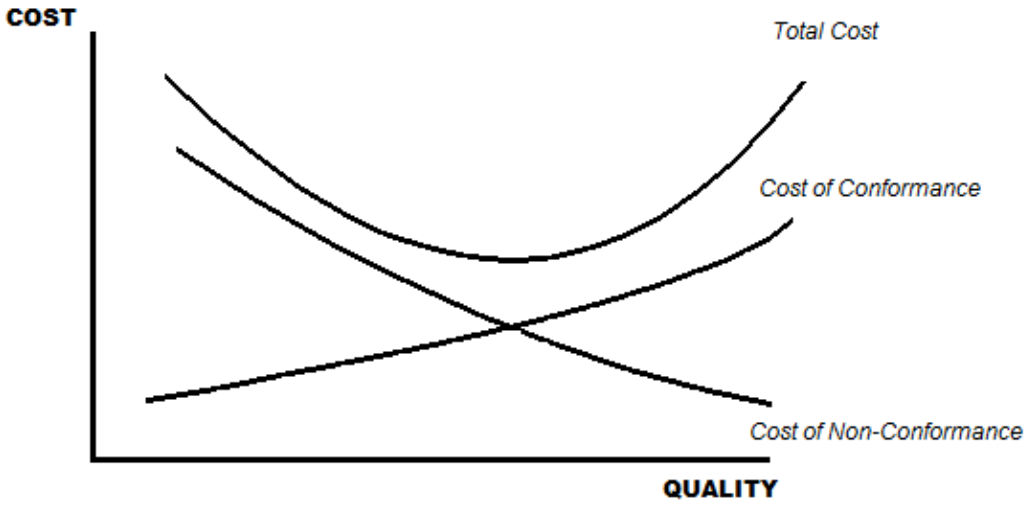

Figure 4: Relationship between cost of conformance and cost of non-conformance

It is expected that various types of quality assurance mechanisms have different ratios of cost of conformance and cost of non-conformance. Since non-conformance may result in lost reputation and profit to an unknown extent, it is considered the more risky portion, thus the goal is to minimize non-conformance. Utilization of additional quality assurance techniques would cause

the cost of non-conformance to decrease, while obviously increasing the costs of conformance (Figure 4). Thus, in order to optimize quality costs, analyzing conformance and non-conformance costs is imperative.

4. CROWDSOURCING COST MODELS

This section presents the cost models for the groups of quality assurance mechanisms which were introduced in Section 2 of this technical report. Some of the definitions of common terms among the cost models are provided to increase the understandability. **Direct cost** is the cost of one task, without any redundancy, controlling party or utilization of any other quality assurance mechanism. Thus total direct cost is the total cost of the job only when assuming that all the tasks are performed in perfect quality and no measures are needed to assure the quality. **Conformance costs** are the total costs occur while appraising the quality of submissions and preventing low quality contributions. This includes all the costs of introducing and maintaining quality assurance mechanisms. **Non-conformance** costs occur when the submissions do not comply with the quality requirements. This includes fixing or replacing the submissions with low quality and all other costs incurred due to poor quality which are unaccounted for.

It should be noted that many related studies have focused on crowdsourcing costs using probabilistic approaches (Sheng, 2008), (Ipeirotis, 2010), (Hirth, 2013). Using a probabilistic cost model will result in less precise yet more accurate estimations. However this is only possible if the probabilities are known or can be estimated. Better probability estimations will result in more accurate cost estimations. The probability of a submission's correctness depends on many parameters, including the characteristics of the crowd, the nature of work and the incentives. Since it is an emerging field, crowdsourcing researchers do not yet have adequate data to form generalizable probabilistic models covering all these parameters. In these cost models the probabilities are included as variables. Thus, $\mathbf{P_w}$ value is the probability of workers to submit a contribution with poor quality either because they deliberately try to cheat the system or make an honest mistake.

The goal of any quality assurance mechanism is either to prevent or detect low quality submissions. Those of which are detected by the quality assurance mechanisms are referred to as *internal failures* and assumed to cause rework in order to complete the work product complying with the quality criteria. The low quality submissions which cannot be detected or prevented by the quality assurance mechanism are passed on to the end product, potentially resulting in

external failures, cheater attraction and faults in trust based systems. The impact of the negative effect of external failures is difficult to estimate. Thus it is imperative to decrease the occurrences of external failures.

In order to achieve a complete end product, it is assumed that all outputs which fail to comply with the quality criteria need to be replaced. Therefore, internal failures cause **rework** and **retest**. If poor quality outputs which cannot be detected by the quality assurance mechanisms are placed as a part of the end product, external failures may occur. The results of these failures are often difficult to represent with monetary costs, such as impacts on business continuity, failure to achieve goals, damage occurrence, or even customer loss. In this technical report such costs are represented as $\mathbf{C_{err}}$. C_{err} largely depends on the end product and the business domain in which the product is to be used.

When quality assurance mechanisms fail to distinguish between poor quality and good quality contributions, long term problems may arise regarding the trust mechanisms and crowd behavior. If workers' good quality submissions are being denied frequently by the quality assurance mechanisms, the workers may change their behavior and cease completing tasks in good faith. Similarly, if the cheaters observe that their poor quality contributions are often being accepted they are encouraged to continue cheating. The damage done to the worker community and reputation and trust mechanisms are denoted as $\mathbf{C_{dmg}}$. C_{dmg} is a global variable and currently there is no way to estimate or control this type of damage and its long lasting, large spectrum effects. However this does not mean that it should be ignored. A good practice is to use C_{dmg} as a risk / cost adjustment factor within the cost of quality calculations.

The possible outcomes of using a quality assurance mechanism are described in Figure 5. The quality assurance mechanism may decide that the contribution complies with the quality criteria identifying it as **positive**. If the quality assurance mechanism decides that the contribution fails to achieve the quality criteria, it identifies the contribution as **negative**. In some cases the quality assurance mechanism can fail to reach a conclusion whether or not the contribution fits the quality criteria. In those cases the outcome is **inconclusive**.

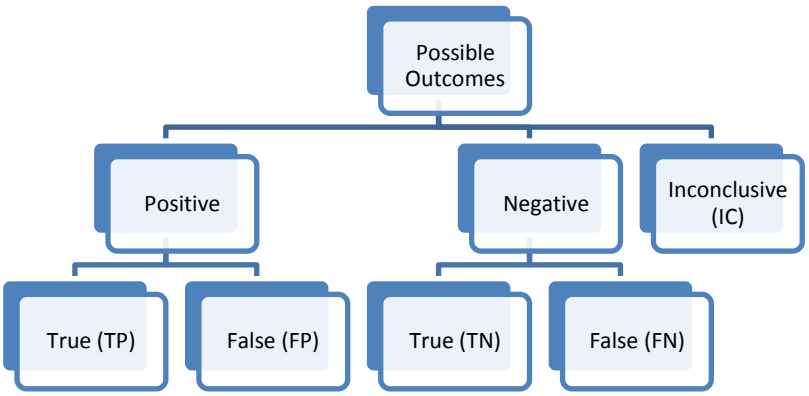

Figure 5: Possible outcomes of a generic quality assurance mechanism

It is possible that quality assurance mechanisms reach wrong conclusions about the contributions, resulting in False Positive (FP) and False Negative (FN) outcomes. When the work output complies with the quality criteria and the quality assurance mechanism results in accepting the output, it is considered a True Positive (TP) decision. In a TP situation there is no additional cost. The contribution is accepted and will not cause any quality related problem in the future.

When the quality assurance mechanism correctly decides the output is a negative match with the quality criteria, it is True Negative (TN). In this case the quality assurance mechanism accomplishes its purpose of detecting a low quality work item. However the item maybe needs to be discarded and rework is needed. The newly produced item will also be subject to quality control and obviously these result in additional costs and delays in task completion.

The outputs which the quality assurance mechanism incorrectly rejects are FN and those of that are incorrectly accepted are FP. Both FN and FP are the undesirable outcomes of quality assurance mechanisms and have negative effect on the quality of the overall work.

4.1 Redundancy

The redundancy quality assurance process can produce 3 possible outcomes. Redundancy does not explicitly deny an output but rather assumes selecting the output with better perceived quality. Thus, the output is placed among the end product whether it fits the quality criteria or not. The only exception is the inconclusive outcome.

Figure 6 shows the potential outcomes of redundancy quality assurance mechanisms. When an output is selected among a few other outputs produced by different instances of the same microtask, it is assumed to be of high quality. The probability of redundancy quality assurance process selecting the output with truly high quality is P_{TP}. P_{FP} is the probability of the quality assurance mechanism to fail to filter out poor quality output and potentially erroneous output is placed among the end product. With the probability of P_{IC} the redundancy quality assurance mechanism fails to achieve a conclusion about the quality of the submission. Inconclusive outcome can happen when none of the outputs of different instances of the same microtask can be selected. For example, if the number of redundant instances (**m**) is even, and the votes are in balance then a consensus cannot be reached.

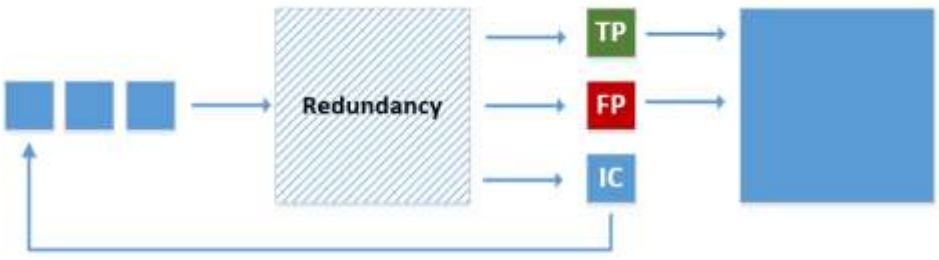

Figure 6: Possible outcomes of redundancy quality assurance mechanisms

Direct cost:

Direct cost of any microtask is assumed to be C_0. The end product consists of outputs produced as a result of **N** microtasks.

[1] C_0: cost of completing one instance of a microtask

Cost of conformance (CoC_{Red}):

[2] $CoC_{Red} = N \,.\, ((m - 1) \,.\, C_0 + C_{agg})$

The conformance cost of redundancy quality assurance mechanisms are mainly caused by the repeated work and output aggregation. The cost of completing one instance of a microtask is assumed to be C_0 [1]. Completing **m** multiple instances of a single microtask as a means of assuring quality increases the costs (**m-1**) times C_0 plus the costs of aggregation (C_{agg}).

In contrast to other quality assurance mechanisms, in redundancy, rework only occurs when the outcome is *inconclusive*. However, redundant production is similar to rework in principle. Every time a redundant instance of a microtask is performed regardless of the outcome of the prior outputs produced by another instance of the same microtask, rework occurs. This rework is represented in the formula [2] by the part: $\mathbf{C_0 \,.\, (m\text{-}1)}$. The probability of redundancy (rework in advance) is **1**, thus omitted.

Cost of selecting the best (or most frequent) output is $\mathbf{C_{agg}}$. It depends on the aggregation method and the number of microtask instances: **m**. When aggregation is done automatically then it does not increase the costs, but if a control group technique is used for aggregation the aggregation costs should be calculated by using control group cost models.

Cost of non-conformance ($CoNC_{Red}$):

[3] $CoNC_{Red} = C_{IF} + C_{EF}$

[4] $C_{IF} = N \,.\, P_{IC} \,.\, (m \,.\, C_0 + C_{agg})$

[5] $C_{EF} = N \,.\, P_{FP} \,.\, (C_{err} + C_{dmg})$

The cost of non-conformance ($CoNC_{Red}$) is the sum of cost of internal failures (C_{IF}) and external failures (C_{EF}), as shown in formula [3].

Internal failure costs are the costs that emerge when the quality assurance mechanism detects a non-conformance and as a result rework and retest occurs. In redundancy quality assurance mechanisms, such detection does not occur, because the selected output is always assumed to be the one with the best perceived quality. However rework and retest can occur when the redundancy quality assurance mechanism reaches to an inconclusive state. Inconclusive state is reached with the probability of P_{IC} and when this state is reached it causes rework and retest of all instances of that particular microtask [4].

External failures occur when a poor quality output is accepted and placed among the end product. The probability of an external failure is P_{FP}. External failure leads to potential error in the end product (C_{err}) and damage done to the reputation and trust mechanisms and the worker community (C_{dmg}). Cost of external failure is difficult to estimate.

Finally the CoQ of redundancy quality assurance mechanisms is expressed with the formula [6].

[6] $CoQ_{Red} = N \,.\, (((m-1) \,.\, C_0 + C_{agg}) + P_{IC} \,.\, (m \,.\, C_0 + C_{agg}) + P_{FP} \,.\, (C_{err} + C_{dmg}))$

4.2 Control group

Control group quality assurance process has 4 possible outcomes. Figure 7 shows these outcomes and the control group quality assurance process. P_{TP} is the probability of the worker submitting a high quality output and the control group correctly decides that it is valid. P_{TN} is the probability of the worker making a poor quality submission and the control group correctly decides that it is invalid. P_{FP} is the probability of control group accepting a poor quality contribution and P_{FN} is the probability of control group to deny a good quality contribution by mistake.

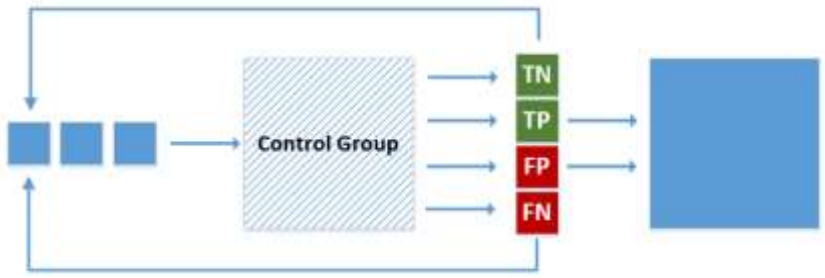

Figure 7: Possible outcomes of control group quality assurance mechanisms

Direct cost:

Direct cost of any task is assumed to be C_0 and the cost of controlling the outputs of one task is C_1.

[1] C_0: cost of completing one instance of a microtask

Cost of conformance (CoC_{CG}):

[2] $CoC_{CG} = N \cdot C_1$

The conformance costs are caused by the additional control tasks. Generally controlling outputs of a microtask is significantly less complex and thus is less expensive. Formula [2] assumes that it costs C_1 to control an output of one microtask. It should be noted that if the output of one microtask is controlled by multiple control group workers, it contains redundancy. Then redundancy cost models should be applied as well. An example is shown in the formula variation [2a]. The C_1 should be placed in [2] for calculation.

[2a] $C_1 = m \cdot C_{ctrl} + C_{agg}$

In [2a] cost of one instance of a control task is C_{ctrl} and an output is controlled **m** times. The aggregation costs are C_{agg}.

Cost of non-conformance ($CoNC_{CG}$):

[3] $CoNC_{CG} = C_{IF} + C_{EF}$

[4] $C_{IF} = N \cdot (P_{FN} + P_{TN}) \cdot (C_0 + C_1)$

[5] $C_{EF} = N \cdot ((P_{FP} + P_{FN}) \cdot C_{dmg} + P_{FP} \cdot C_{err})$

The cost of non-conformance ($CoNC_{Red}$) is the sum of cost of internal failures (C_{IF}) and external failures (C_{EF}), as shown in formula [3].

Internal failure costs are caused by rework and retest which occur when the quality assurance mechanism detects a non-conformance. When the controlling workers decide that the submission does not comply with quality criteria, the output of the task is denied and rework and retest is needed to produce the same output. Control group either identifies poor quality work correctly or incorrectly giving the probability of a work output to be denied as $P_{FN} + P_{TN}$. The cost of rework and retest is $C_0 + C_1$ as shown in the formula [4].

An erroneous work output can be placed among the end product only if the control group incorrectly decides it is valid. The costs occur when an external failure occurs in the end product are denoted as C_{err}. [5].

Whether the control group fails to detects a poor quality submission (P_{FP}) or else identifies a good quality output of a microtask as invalid (P_{FN}) damages occur to the trust mechanisms and worker community (C_{dmg}). [5].

Finally the CoQ of control group quality assurance mechanisms is expressed with the formula [6].

[6] $\text{CoQ}_{CG} = N\,.((C_1) + (P_{FN} + P_{TN})\,.(C_0 + C_1) + (P_{FP} + P_{FN})\,.\,C_{dmg} + P_{FP}\,.C_{err})$

4.3 Gold standard

Direct cost of any task is assumed to be C_0.

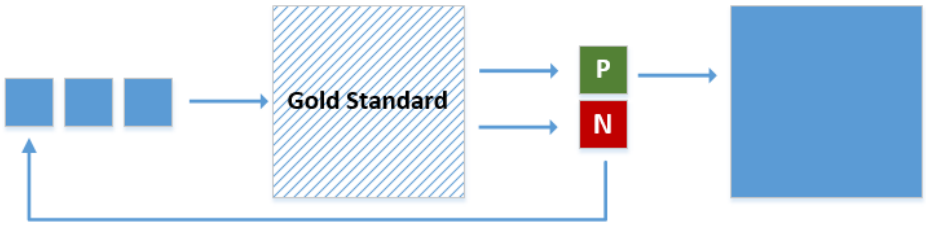

Figure 8: Possible outcomes of gold standard quality assurance mechanisms

As described in Section 2.3, gold standard quality assurance mechanism can be used in different and usage scenarios can be grouped as synchronous and asynchronous usage. In synchronous usage, gold standard tasks are provided to the user along with a number of normal tasks. In this case the decision to approve or deny the submissions is based on the comparison of the gold standard output and the predefined expected result. If the gold standard output is valid then the entire group of task outputs is accepted. Possible outcomes of a gold standard quality assurance mechanism are shown in Figure 8.

In asynchronous usage gold standard, tasks are assigned either in the beginning or randomly among a series of other tasks. This type of gold standard usage is generally for qualifying or training the worker or trying to identify submission pattern of the worker, supporting a reputation system.

Direct cost:

[1] C_0: cost of completing one instance of a microtask

Cost of conformance (CoC_{GS}):

C_{exp}: cost of introducing one gold standard task.

X: sample size in gold standard pool

[2a] $$CoC_{GS} = X \cdot C_{exp} + N \cdot \frac{k}{t-k} \cdot C_0$$

[2b] $$CoC_{GS} = X \cdot C_{exp} + m \cdot \frac{k}{t-k} \cdot C_0$$

Formula [2a] shows the conformance costs for synchronous usage of gold standard quality assurance mechanisms where (k / t – k) is the ratio of number of gold standard tasks to the number of normal tasks which are assigned together.

Formula [2b] represents the conformance costs for the asynchronous usage where **m** gold standard tasks are assigned in the beginning of the task sequence either for training or qualification of the worker.

CoC_{GS} also includes costs of introducing the gold standard tasks. This is generally done by an expert and usually expert tasks costs more than normal microtasks. Developing X gold standard data and inserting them to the system costs $X \cdot C_{exp}$.

Cost of non-conformance ($CoNC_{GS}$):

[3] $$CoNC_{GS} = C_{IF} + C_{EF}$$

[4] $$C_{IF} = N \cdot \frac{k}{t-k} \cdot (1 - (P_P)^k) \cdot (t - k + k) \cdot C_0$$

[5] $$C_{EF} = N \cdot \frac{k}{t-k} \cdot (P_P)^k \cdot P_W \cdot (t - k) \cdot (C_{err} + C_{dmg})$$

Non-conformance cost ($CoNC_{GS}$) is the sum of internal failure costs (C_{IF}) and external failure costs (C_{EF}), as shown in formula [3].

Internal failure costs occur when the quality assurance mechanism detects a nonconformance and process results in rework and retest. In synchronous gold standard quality assurance process, a group of microtask outputs are associated with one or more gold standard tasks. These outputs are either accepted or denied by comparing the submission made for the gold standard task and the expected result. If the submission varies from the expected result, the associated tasks are also denied. Formula [4] represents the costs of an internal failure. **k** is the number of gold standard tasks in the task batch of a total size of **t**, including k gold standard tasks. $(\frac{k}{t-k})$ represents the ratio of gold standard tasks to the normal tasks. $1\text{-}P_P$ is the probability of a worker to submit a poor quality (negative) contribution to a gold standard task. (t – k) is the number of normal tasks to be denied and needs to be reworked on. Using a gold standard quality assurance mechanism to retest (t – k) normal tasks requires k gold standard tasks. When added, the number of microtasks to be reworked and retested is **t**.

External failure costs are caused when the quality assurance mechanism fails to detect a poor quality contribution. In synchronous gold standard quality assurance process, if the worker makes a valid submission for the gold standard tasks in a batch but provides poor quality contributions for normal tasks in that batch, external failures become possible. This case is represented by the formula [5]. P_P is the probability of a user to submit a valid contribution for a gold standard task and P_W is the probability of a user to make a poor quality submission for one normal task. (t – k) represents the number of normal tasks in a batch, and when multiplied by the false submission probability, it gives the number of poor quality outputs in a batch of microtasks.

As stated before external failures caused by FP submissions lead to both a potential error in the end product and damage done to the trust mechanisms and the worker community by attracting cheaters. Both cases are represented in the formula [5]. External failures which are caused by FN submissions also damage the worker community and the trust system. However FN in synchronous gold standard is a different case than other quality assurance mechanisms. In other quality assurance mechanisms, if the mechanism rejects a high quality submission, when there is no wrongdoing on the worker's side, the worker may feel discouraged to make honest and careful submissions. On the other hand, in gold standard the workers' submissions are denied only if the worker provides a low quality output to a gold standard task. Since there is a mistake in worker's side, it is debatable if FN decisions damage the worker community or not.

The complete formula of a gold standard quality assurance mechanism is displayed in [6].

[6] $\mathrm{CoQ_{GS}} = X \,.\, \mathrm{C_{exp}} + N \,.\left(\frac{k}{t-k}\right) .(C_0 + (1-(P_P)^k) \,.\, \mathrm{t} \,.\, C_0 + (P_P)^k .\; P_W \,.\, (\mathrm{t}-\mathrm{k}).\left(C_{err} + C_{dmg}\right))$

Since in asynchronous gold standard usage, gold standard tasks are not associated with normal tasks, denying a gold standard task output does not necessarily result in rework or external failures. Thus cost models for asynchronous gold standard should be derived on case basis according to the design of the quality assurance mechanism.

The cost models provided for gold standard quality assurance mechanisms greatly depend on the probability of a worker to submit correct and incorrect results for both gold standard and normal tasks. These probability values are not only related with the characteristics of the worker community but also the design of the gold standard quality

assurance mechanism. Thus, it is important to design the gold standard tasks in a way that the probability of a totally random submission to be true is as low as possible.

4.4 Worker centric & design centric

The cost of quality of worker and design centric quality assurance mechanisms greatly depend on the particular use cases of those mechanisms. It may not provide accurate insights to develop a generalized model for these types of quality assurance mechanisms. However, it is possible and advised to derive specific cost models by using the same approach presented and exemplified in this technical report.

5. DISCUSSION

Foreseeing the impact of quality assurance optimization research: Coping with inefficiencies at a global scale

Applying quality assurance mechanisms raises costs significantly. When these mechanisms are used excessively or incorrectly, inefficiencies occur which result in vast amount of wasted effort in global scale. At the time this technical report is published, approximately 2.500 jobs which in total contain more than 1.500.000 human intelligence tasks (HITs) were posted on Amazon Mechanical Turk. Without doubt many more are being crowdsourced on other large scale crowdsourcing platforms as well. These numbers indicate that crowdsourcing has a massive usage. Since conformance costs have significantly higher ratio in crowdsourcing compared to traditional way of production, even small improvements in efficiency result in huge savings. The cost models presented in this technical report can be used simply to select quality assurance mechanisms which fit the job better or design efficient hybrid quality assurance mechanisms. We foresee that by enabling savings at microtask levels it is possible to make a significant impact on crowdsourcing efficiency at a global scale.

Long lasting effects

When analyzing the costs of potential outcomes of quality assurance mechanisms we considered the costs of damages done to the worker community and trust mechanisms. We understand that these cost values may not be estimated accurately. However it is important for the crowdsourcing practitioners to understand the long lasting side effects and indirect costs of the quality assurance mechanisms they use, in order to enable crowdsourcing as a sustainable means of production.

Real life usage of cost of quality models

The cost models introduced in this technical report can be used to estimate the costs that occur according to the quality assurance mechanism selection or design. The cost models include probabilistic parameters. These parameters depend on various characteristics such as the crowd, nature of work and incentive mechanisms. Crowdsourcing practitioners can use simulations to calculate cost estimations, which may guide them to make better quality assurance mechanism selections or designs. The more realistic probability values are used, the more accurate estimations can be done. Thus crowdsourcing practitioners are advised to make observations of crowd behavior and the effects of design decisions on this behavior, and use the observed probabilistic values as parameters with the cost of quality models.

6. REFERENCES

Antin, J., & Shaw, A. (2012). Social desirability bias and self-reports of motivation: a study of amazon mechanical turk in the US and India. InProceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 2925-2934). ACM.

Callison-Burch, C. (2009). Fast, cheap, and creative: evaluating translation quality using Amazon's Mechanical Turk. In Proceedings of the 2009 Conference on Empirical … (pp. 286–295). Retrieved from http://dl.acm.org/citation.cfm?id=1699548

Crosby, B. P. (1979). Quality is free: The Art of Making Quality Certain McGraw-Hill. New York.

Difallah, D. E., Demartini, G., & Cudré-Mauroux, P. (2013). Pick-a-crowd: tell me what you like, and i'll tell you what to do. In Proceedings of the 22nd international conference on World Wide Web (pp. 367-374). International World Wide Web Conferences Steering Committee.

Eagle, N. (2009). txteagle: Mobile crowdsourcing. In Internationalization, Design and Global Development. Retrieved from http://link.springer.com/chapter/10.1007/978-3-642-02767-3_50

Hirth, M., Hoßfeld, T., & Tran-Gia, P. (2011). Cost-Optimal Validation Mechanisms and Cheat-Detection for Crowdsourcing Platforms. In 2011 Fifth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (pp. 316–321). Ieee. doi:10.1109/IMIS.2011.91

Hirth, M., Hoßfeld, T., & Tran-Gia, P. (2013). Analyzing costs and accuracy of validation mechanisms for crowdsourcing platforms. Mathematical and Computer Modelling, 57(11-12), 2918–2932. doi:10.1016/j.mcm.2012.01.006

Ho, C. J., & Vaughan, J. W. (2012). Online Task Assignment in Crowdsourcing Markets. In AAAI.

Hossfeld, T., Hirth, M., Tran-Gia, P.; (2011). Modeling of Crowdsourcing Platforms and Granularity of Work Organization in Future Internet. In International Teletraffic Congress. San Francisco, USA

Huang, E., Zhang, H., Parkes, D. C, Gajos, K.Z, Chen, Y.: (2010). Toward automatic task design: A progress report. In Proceedings of the ACM SIGKDD workshop on human computation. ACM, 77–85.

Ipeirotis, P. G., Provost, F., & Wang, J. (2010). Quality management on Amazon Mechanical Turk. Proceedings of the ACM SIGKDD Workshop on Human Computation - HCOMP '10, 64. doi:10.1145/1837885.1837906

Karger, D. R., Oh, S., & Shah, D. (2011). Budget-optimal crowdsourcing using low-rank matrix approximations. 2011 49th Annual Allerton Conference on Communication, Control, and Computing (Allerton), 284–291. doi:10.1109/Allerton.2011.6120180

Kazai, G., Kamps, J., & Milic-Frayling, N. (2011). Worker types and personality traits in crowdsourcing relevance labels. Proceedings of the 20th ACM international conference on Information and knowledge management - CIKM '11, 1941. doi:10.1145/2063576.2063860

Kern, R., Thies, H., Bauer, C., & Satzger, G. (2010). Quality assurance for human-based electronic services: A decision matrix for choosing the right approach. In Current Trends in Web Engineering (pp. 421–424). Retrieved from http://link.springer.com/chapter/10.1007/978-3-642-16985-4_39

Kittur, A, & Kraut, R. (2008a). Harnessing the wisdom of crowds in wikipedia: quality through coordination. In Proceedings of the 2008 ACM conference on …. Retrieved from http://dl.acm.org/citation.cfm?id=1460572

Kittur, A, Chi, E. H., & Suh, B. (2008b). Crowdsourcing user studies with Mechanical Turk. In Proceeding of the twenty-sixth annual CHI conference on Human factors in computing systems - CHI '08 (p. 453). New York, New York, USA: ACM Press. doi:10.1145/1357054.1357127

Kittur, A, Smus, B., Khamkar, S., & Kraut, R. (2011). Crowdforge: Crowdsourcing complex work. In Proceedings of the 24th annual …. Retrieved from http://dl.acm.org/citation.cfm?id=2047202

Le, J., & Edmonds, A. (2010). Ensuring quality in crowdsourced search relevance evaluation: The effects of training question distribution. In … crowdsourcing for search … (pp. 17–20). Retrieved from http://ir.ischool.utexas.edu/cse2010/materials/leetal.pdf

McCann, R., Shen, W., & Doan, A. (2008). Matching Schemas in Online Communities: A Web 2.0 Approach. 2008 IEEE 24th International Conference on Data Engineering, 110–119. doi:10.1109/ICDE.2008.4497419

Okubo, Y., Kitasuka, T., & Aritsugi, M. (2013). A Preliminary Study of the Number of Votes under Majority Rule in Crowdsourcing. Procedia Computer Science, 22, 537–543. doi:10.1016/j.procs.2013.09.133

Oleson, D., Sorokin, A., Laughlin, G., & Hester, V. (2011). Programmatic Gold: Targeted and Scalable Quality Assurance in Crowdsourcing. Human computation, 43–48. Retrieved from http://www.aaai.org/ocs/index.php/WS/AAAIW11/paper/viewFile/3995/4267

Paiement, J.F, Shanahan, J.G., Zajac, R.; (2010). Crowdsourcing local search relevance. Proceedings of the CrowdConf 2010.

Quinn, A., & Bederson, B. (2011). Human computation: a survey and taxonomy of a growing field. In … Conference on Human Factors in Computing …. Retrieved from http://dl.acm.org/citation.cfm?id=1979148

Rogstadius, J., Kostakos, V., Kittur, A., & Smus, B. (2011). An Assessment of Intrinsic and Extrinsic Motivation on Task Performance in Crowdsourcing Markets. In ICWSM (pp. 321–328). Retrieved from http://www.aaai.org/ocs/index.php/ICWSM/ICWSM11/paper/viewFile/2778/3295

Ross, J., Irani, L., & Silberman, M. (2010). Who are the crowdworkers?: shifting demographics in mechanical turk. In CHI'10 Extended … (pp. 2863–2872). Retrieved from http://dl.acm.org/citation.cfm?id=1753873

Schiffauerova, A., & Thomson, V. (2006). A review of research on cost of quality models and best practices. International Journal of Quality & Reliability Management, 23(6), 647–669. doi:10.1108/02656710610672470

Shaw, A. D., Horton, J. J., Chen, D. L.; (2011). Designing incentives for inexpert human raters. In Proceedings of the ACM 2011 conference on Computer supported cooperative work. ACM, 275–284.

Sheng, V., Provost, F.; Ipeirotis, P. G.; (2008). Get another label? improving data quality and data mining using multiple, noisy labelers. In Proceedings of the 14th ACM SIGKDD international

Snow, R., O'Connor, B., Jurafsky, D., & Ng, A. (2008). Cheap and fast---but is it good?: evaluating non-expert annotations for natural language tasks. In … methods in natural language …. Retrieved from http://dl.acm.org/citation.cfm?id=1613751

Sorokin, A., & Forsyth, D. (2008). Utility data annotation with amazon mechanical turk. In Computer Vision and Pattern … (pp. 1–8). Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4562953

Stewart, O., Lubensky, D., & Huerta, J. (2010). Crowdsourcing participation inequality: a SCOUT model for the enterprise domain. In … of the ACM SIGKDD Workshop on … (pp. 30–33). Retrieved from http://dl.acm.org/citation.cfm?id=1837895

Stvilia, B., & Twidale, M. (2008). Information quality work organization in Wikipedia. … society for information …, 59(6), 983–1001. doi:10.1002/asi

Von Ahn, L., & Dabbish, L. (2004). Labeling images with a computer game. Proceedings of the 2004 conference on Human factors in computing systems - CHI '04, 319–326. doi:10.1145/985692.985733

Von Ahn, L., & Dabbish, L. (2008). Designing games with a purpose. Communications of the ACM, 51(8), 57. doi:10.1145/1378704.1378719

Voyer, R., & Nygaard, V. (2010). A hybrid model for annotating named entity training corpora. In … Linguistic Annotation … (pp. 243–246). Retrieved from http://dl.acm.org/citation.cfm?id=1868759

Vukovic, M. (2009). Crowdsourcing for Enterprises. In Congress on Services - I (pp. 686–692). doi:10.1109/SERVICES-I.2009.56

Welinder, P., & Perona, P. (2010). Online crowdsourcing: Rating annotators and obtaining cost-effective labels. 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops, 25–32. doi:10.1109/CVPRW.2010.5543189

Xia, T., Zhang, C., Xie, J., & Li, T. (2012). Real-time quality control for crowdsourcing relevance evaluation. 2012 3rd IEEE International Conference on Network Infrastructure and Digital Content, 535–539. doi:10.1109/ICNIDC.2012.6418811

Zhai, Z., Hachen, D., Kijewski-Correa, T., Shen, F., & Madey, G. (2012). Citizen Engineering: Methods for "Crowdsourcing" Highly Trustworthy Results. 2012 45th Hawaii International Conference on System Sciences, 3406–3415. doi:10.1109/HICSS.2012.151