Quality Control in Crowdsourcing Systems

Mohammad Allahbakhsh[1] Boualem Benatallah[1]
Hamid Reza Motahari-Nezhad[1,2] Aleksandar Ignjatovic[1]
Elisa Bertino[3]

[1] University of New South Wales
Sydney 2052, Australia
{mallahbakhsh,boualem,ignjat}@cse.unsw.edu.au

[2] HP Labs Palo Alto, CA, USA
hamid.motahari@hp.com

[3] Purdue University, West Lafayette, Indiana, USA
bertino@cs.purdue.edu

Technical Report
UNSW-CSE-TR-201205
February 2012

THE UNIVERSITY OF
NEW SOUTH WALES

School of Computer Science and Engineering
The University of New South Wales
Sydney 2052, Australia

Abstract

Crowdsourcing as a new model of distributed computing enables people to leverage the intelligence and wisdom of the crowd toward solving problems. Quality control is a critical aspect in crowdsourcing. This article proposes a framework for characterizing various dimensions of the quality control in crowdsourcing systems. We also review some existing crowdsourcing systems and identify open issues.

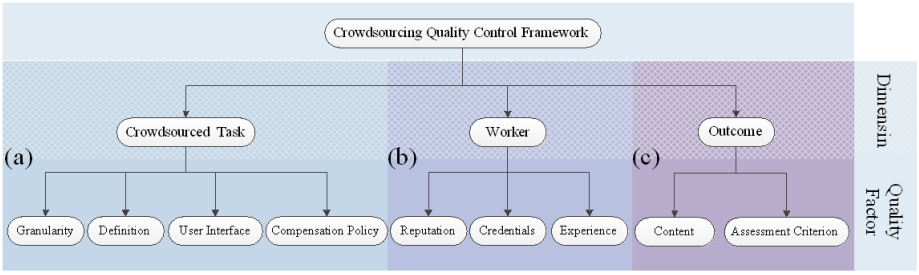

Figure 1.1: Crowdsourcing Quality Control Framework

1 Introduction

Crowdsourcing involves receiving, incorporating and consolidating contributions from a large crowd with varied levels of expertise [7]. Processes like building artifacts, evaluating items, sharing information and executing tasks are some instances of crowdsourcing processes [5]. For instance, Amazon Mechanical Turk[1] provides on-demand access to task forces for micro-tasks such as image recognition, language translation, etc. Several organizations including DARPA and various world health and relief agencies are using platforms such as Amazon Mechanical Turk (MTurk) and Ushahidi[2] to crowdsource information through multiple channels, including SMS, email, Twitter and the Web in general. To crowdsource a task, task owners, also called the *Requesters*, submit tasks to a crowdsourcing platform. Another group of people, called *workers*, can contribute to solving tasks (the result of solving a task is called also *outcome*). Requesters evaluate the outcomes and may reward workers whose outcomes have been accepted [12].

Characterizing quality in crowdsourcing is challenging. Quality in various areas of computing including software engineering, data management and open source is defined using multifaceted models in which attributes like reliability, accuracy, relevancy, completeness, and consistency contribute [10]. In our work, we adopt the general Crosby's definition of quality [3] as guide to identify quality control attributes including dimensions and factors. This definition emphasize "conformance to requirements" as a guiding principle to define quality control models. The overall quality of a crowdsourced task depends on *the task* which is being crowdsourced, *the workers* who contribute to solving the task, and *outcomes* of work. It should be noted that, issues such as quality of services like reliability and security, while very important to the success of crowdsourcing platforms, are outside the scope of this paper as they are related to software engineering aspects of platforms.

In this article, we propose a conceptual framework characterizing various dimensions of quality control in crowdsourcing systems. The proposed framework is illustrated in Figure 1.1 and described in next sections. We also leverage this framework to survey and analyze quality control techniques in some representative crowdsourcing prototypes and platforms. We conclude with a discussion on open issues in quality control in crowdsourcing systems.

[1] www.mturk.com
[2] http://ushahidi.com

2 Crowdsourced Task

The *crowdsourced task* is the problem offered to be solved by the crowd. We identify four important factors contributing to the quality regarding this dimension: *task granularity*, *definition*, *user interface* and finally *compensation policy*. Task dimension and its related quality factors are depicted in Figure 1.1(a).

2.1 Granularity

In terms of granularity, tasks can be divided into two broad types, *simple tasks* and *complex tasks*. Simple tasks are the self-contained, appropriately short tasks which usually need a little expertise to be solved [12]. Human Intelligence Tasks (HITs) in MTurk are instances of simple tasks. Annotating photos, writing a short description for an image or transcription of a recorded audio are samples of commonly used HITs in MTurk. On the other hand, solving a complex task like writing an article is not as simple as posting a simple task to a crowdsourcing system. Solving such a complex problem needs more time, costs and expertise, so few people will be interested in doing it. The popular solution for these tasks is following the general map-reduce model [12]. In this model the complex task is divided into some smaller simple sub-tasks (map step); sub-tasks are submitted to the crowdsourcing platform and crowd asked to accomplish them and finally the outcomes of sub-tasks are consolidated to build the final answer of the complex task (reduce step). The workflow of such a complex task is how these simple sub-tasks are chained together to build the general complex task [2]. The workflow of a complex task can be iterative [14, 13], parallel [14], or a combination of these models [12, 2, 14, 1].

Designing workflow for complex tasks greatly affects the quality of the task outcome [12, 2, 1]. A poor workflow design can lead to low quality results. For instance, it is shown in [12] that designing a poor outline for an essay to be written by the crowd may result in a low quality essay; also, improving the quality of an outline using crowd contribution increases the quality of the corresponding written essays.

2.2 Task Definition

Task definition is the information provided by the requester to potential workers making them aware of the details of the task which is advertised for crowdsourcing. The details mainly consist of two items. The first item is a short description of the task, explaining the nature of the task, time limitations, etc. The second item is qualification requirements of the task. These qualifications specify the eligibility criterion by which workers will be evaluated before being accepted for participation in the crowdsourced task. For example, in MTurk, requesters can specify that only workers having percentage of accepted works larger than 90% can participate, or that only the workers living in US can participate in a particular survey. Studies show that quality of the provided definition for a task impacts the quality of the outcomes [9].

2.3 Task User Interface

The *task interface* design refers to the interface through which the workers access the task. Interface can be a Web interface, an API or any other kind of user interface. The user interface must simplify contribution of the workers to the task [5]. A user friendly interface can attract more workers and increases the chance of receiving a high quality outcome [9]. The negative side is that a simple interface also makes it easier for deceptive workers to exploit the system. On the other hand, an unnecessarily complicated interface will discourage honest workers and may lead to delay in performing the task. The model in which task user interface is designed so that cheating is not easier than doing the task is in some studies called *defensive design* [9, 15]. In addition to defensiveness, a good interface can help workers to do tasks faster and more accurately [9].

2.4 Compensation Policy

The compensation policy is an important factor impacting the outcome quality. There are two important aspects in choosing a compensation policy: the *amount* of reward and the *rewarding method*. Whereas the reward amount is important and an inappropriate value may result in low quality contributions or even task starvation, increasing the reward amount does not necessarily increase the quality of the outcome. Intrinsic incentives of the participants, such as personal enthusiasm or altruism, in conjunction with the extrinsic ones, such as monetary reward, can motivate honest users to participate in the task. Studies have even shown that in some cases positive effect of intrinsic incentives on the quality of the outcome is more significant than the impact of the extrinsic incentives [15]. Concerning the rewarding method, sometimes the payment method has a bigger impact on the quality of the outcome than the payment amount [9, 15]. For example, in a task requested for finding 10 words in a puzzle, paying-per-puzzle will lead to more solved puzzles than paying-per-word [12].

3 The Worker

Quality of a worker is the likelihood of receiving a satisfying outcome form the worker. We characterize the quality of a worker using *reputation*, *credentials*, and *experience* factors (See Figure 1.1(b)).

3.1 Reputation

The *reputation* refers to the community-wide judgment on the capabilities of a worker [6]. In terms of the information sources used for calculating the reputation, reputation can be categorized in two categories [4]: *content-based reputation* and *feedback-based reputation*. Content-based reputation ranks are calculated on the basis of the quality of the outcomes generated by the worker. For example, in the WikiTrust, a tool for checking the quality of Wikipedia[1] articles, the reputation of the worker is calculated based on the quality of the changes she makes to the content. If the change she has made is preserved by consequent editors, she will gain reputation; otherwise she loses reputation [4].

[1] www.wikipedia.org

Feedback-based reputation ranks are calculated based on the feedback received from other workers or requesters on the quality of the outcomes provided by the worker. For instance, in Stackoverflow[2], a question and answer web site, a user can ask questions or answer the questions asked by others. Other community members can cast positive or negative votes on the quality of the user's questions or answers. The aggregation of these positive and negative votes is proposed as the reputation rank of the user. Along with feedback, some works consider the trustworthiness of the person who has given the feedback as well as time and credit aspects towards building a more enriched and informative reputation ranking [8]. As the capabilities and trustworthiness of the people may change in the time, the feedback received from a trustworthy person who has recently had an interaction with a worker is more dependable than a feedback received long time ago from a person with low level of trustworthiness. The credit factor is related to the monetary value paid for doing the task. The reputation built by contributing in high credit processes is more reliable than a reputation built on processes with low credits [8].

3.2 Credentials

Credentials are documents or evidence on which the capabilities of a worker regarding a particular crowdsourcing process can be assessed. Information such as academic certificates or degrees, spoken languages or geographical regions which a worker is familiar with (e.g. where she lives or works) can be used as credentials. The requesters may also ask for workers with particular capabilities and credentials [19]. For instance, in MTurk requesters can define qualification tests and ask workers to take them, and only the workers who passed successfully the qualification tests can be selected to do the task.

3.3 Experience

Experience refers to knowledge and skills a worker has gained while working in the system. In most of the systems, such as MTurk and StackOverflow, workers join as novice users with a basic set of capabilities and can become savvier by gaining more experience through performing tasks and benefiting from training [17]. Some research shows that shepherding and supporting workers by providing additional information on request or prior-submission assessments also helps workers learn more, enhance their expertise and experiences and help them contribute to the tasks more productively [17].

4 Outcome

The *outcome* is the result of the contribution of the worker to a crowdsourced task. The quality of an outcome is measured by how the received outcome conforms to the requirements of the requester. Even high quality workers in a well designed task are likely to provide low quality contributions due to, for instance, mistakes or misunderstanding of the requester's expectations. Quality of an outcome is firstly related to the quality of the content of the generated outcome, e.g., a well written text or high quality image. It is also related

[2]Stackoverflow.com

to the assessment criterion used to evaluate the quality of the outcome. The assessment criterion is designed according to the requirements of the requester. For example, the quality expected of a photo being posted on a personal weblog is different from the expected quality of a photo being posted to the web site of a professional photography magazine; a low quality photo may pass the former but fail on the latter. We identify two types of factors regarding the quality of the outcome: *content* and *assessment criterion*. These factors are illustrated in Figure 1.1(c).

4.1 Content

A broad range of outcomes are generated using crowdsourcing tasks. Long-living artifacts like articles, photos, software code and as well as short-term artifacts like votes on an item or solution to a captcha are samples of outcomes received form workers in crowdsourcing systems [5]. The quality of content of an outcome is highly related to the nature of the outcome. For example, for a textual answer provided to a question in a question and answer system, the length of the answer, unique words used in the answer and answer to question length ratio are some of the content related factors which impact the quality of an outcome [10]. It is possible to identify the content factors for other types of artifacts like images, sounds and videos. For instance, ESP game employs the crowd to extract the content factors of the images [18]. Players describe and identify objects in an image, the topic of the image or other, possibly important features.

4.2 Assessment Criterion

Assessment criterion is the criterion by which the workers' outcomes are evaluated. An assessment criterion in its simplest form can be seen as a set of quality metrics with the range of valid values for every metric. For example (reputation of the worker ¿ 90%, length of answer ¿ 10 words) can be an assessment criterion which implies only the answers to a question will be accepted which have the length of at least 10 words and are generated by workers having reputation greater than 90 out of 100. The quality metrics used in assessment factors vary from context to context. For example, the quality metrics used for evaluating an image are different from metrics used for assessing a text.

Various parameters maybe considered while choosing an assessment criterion. The first parameter can be common sense rules. For instance, in a marketplace an item with the price lower than half of the average of prices of similar articles can be considered as a low value item. The second parameter is the rules and regulations of the system. For instance, in Wikipedia, an article which is not supported with enough documents and references is considered as low quality and is likely to be removed. The requirements of the requester presented in task definition is the third important parameter regarding choosing the assessment criterion; while a result might have the requested quality from point of view of one requester, it might be inadequate from the point of view of another requester with a more stringent criteria.

5 Approaches to Quality Control

In this section we propose taxonomy of approaches to quality control.

5.1 Task Management Approaches

In existing crowdsourcing systems the task management approaches can be broadly categorized as *task preparation* and *workflow management*.

Task Preparation

Task preparation comprises of providing three quality factors: an effective task definition, which is an unambiguous description of a crowdsourced task via its specifications and requirements; a user friendly interface, and an appropriate compensation policy for the task. There are no existing approaches which adequately provide all of these three factors; the proposed prototypes provide only partial solutions for some of the factors. For instance, the work proposed in [11] for rating articles, addresses the problem of interface design. The users are asked to provide a rating for every article ranging from 1 to 7. To prevent cheating, users are asked to also provide a short description of the article along with recommendations for improving the text. The work presented in [15] addresses the problem of choosing an adequate compensation policy as means for improving the likelihood of high quality outcome.

Effective task preparation is highly non-trivial. Since it is not possible to provide a general solution for tasks in all contexts, task preparation is highly task specific. Providing an informative description for a task is also highly related to the target crowd who will be doing the task and it needs to take into account crowd attributes such as degrees of expertise, language, etc. Design of an adequate user interface must rely on software engineering design techniques which must address simplicity of use while maximizing its defensiveness. Finally, choosing an appropriate compensation policy needs to consider task properties and requirements, social factors like crowd interests, income level and many other parameters in order to provide a good economical model for the task. Again, currently there are no comprehensive approaches for effective task preparation.

Workflow Management

While execution of simple tasks is simple and straightforward, complex tasks have complicated workflows. Workflow management approaches try to help requester increase the quality of outcome through online management of the workflow of the crowdsourced task, taking into account task definition, compensation policy and granularity. Design of a suitable workflow for the task is addressed in [12, 2, 1]. The Requesters even can ask the crowd to contribute to the design of the workflow [1]. They can monitor the execution of the sub-tasks in real-time by controlling spent time and costs so far, and even start new sub-tasks or stop some running ones on the fly. Also, the requesters can manage the way in which the outcomes are consolidated from the outcomes of sub-tasks and thus prevent low quality outcomes from being integrated into the final result. CrowdForge [12], Turkomatic [1] and CrowdWeaver [2] are samples of tools used this approach to quality control.

5.2 Worker Selection Approaches

In crowdsourcing systems three worker selection approaches are commonly used: *open-to-all*, *reputation-based*, and *credential-based* approaches.

Open-to-All

In the open-to-all model no selection mechanisms are applied and every worker can select the task and contribute outcomes. Wikipedia, Threadless[1] and ESP Game [18] are samples of this group of the crowdsourcing systems. This approach is simple and easy to use but its main disadvantage is that it allows every worker with every level of expertise and trustworthiness to contribute to the task. Therefore, due to possibility of presence of low quality workers, it is more likely that many of the received outcomes from the crowd will be of low quality.

Reputation-Based Approaches

Reputation-based approaches employ the reputation of workers to choose adequate workers for a particular task. These approaches control workers' access to a task by using qualification requirements of the task, described in the task definition. MTurk is an example of a system using reputation for worker selection.

While reputation-based approaches are simple and easy to understand, they are also prone to various types of limitations and attacks. Approaches which use the feed-back based reputation are subject to problems like lack of sufficient information, biased user interests, evaluator dishonesty or collusion. Approaches which employ content-based reputations are to some extents robust against these problems but they are not general enough to be applied to all crowdsourcing areas. For instance, for crowdsourcing tasks in which workers are required to vote on similarity of the objects or look for a particular object (e.g. a particular person in a photo) it is not feasible to use these approaches. Moreover, few of feedback-based or content-based reputation approaches consider time, credit and trustworthiness of the evaluator in the process of reputation calculation [8].

Credential-Based Approaches

The credential-based worker selection approaches select workers to do a task based on the task requirements and credentials the workers have. For instance, in Wikipedia, when a user creates a new article or edits and existing one, the contributions are reviewed by administrators. The administrators are elected by users. Only administrators can certify or reject contributions made by normal user. As another example, in Stackoverflow access of the users to some parts of the systems depends on the reputation and badges (credentials) they have.

Credential-based approaches are most suitable for systems in which users are known, such as network access control systems. However, in crowdsourcing systems in which sometimes users are completely unknown, such approaches are not fully applicable, but using them along with other approaches, such as reputation based approaches, can increase the quality of the outcomes provided by the workers.

[1] www.threadless.com

5.3 Outcome Evaluation Approaches

In crowdsourcing the requesters wish to select, incorporate and consolidate the outcomes with the highest possible quality. The following are most common approaches for outcome evaluation.

5.4 Expert Review

This is a manual quality control approach. This model employs the human intelligence and wisdom to evaluate outcomes. Outcomes received from workers are sent to some domain experts for evaluation. Then, based on the evaluation results received form experts, requester (automatically or manually) accepts or rejects the outcomes [16].This is the model which is used in Wikipedia, as well as for evaluating papers for conferences and journals.

Expert review approach is simple and easy to use and usually leads to results with higher quality but it has some important limitations. First, the cost of employing an expert is much higher than using few non-expert workers to evaluate the work. Second, the time of waiting to receive evaluation results from experts are usually longer than waiting for the crowd results.

5.5 Forced Agreement

This model evaluates outcomes based on the agreement of two partners that are solving the task simultaneously and independently [16]. There are two types of forced agreement: (i) *input agreement* and (ii) *output agreement*. In the Input Agreement model, two workers are given inputs that may or may not be the same. Then they are asked to describe the given input to one another. Based on the received descriptions the workers decide whether or not they are dealing with the same input. If both workers agree on the similarity of the inputs, the system deems that inputs are the same. This model is used in Tag-a-Tune game [16].

In the output agreement model, two or more workers are given a same input and asked to describe the input simultaneously and independently. The answer is accepted just when all workers provide the same description. This model is widely used for image labeling. For instance, ESP Game uses this model for labeling images online [5, 16].

Forced agreement is a fast approach evaluates outcomes instantly, but it is applicable only to a limited range of simple tasks. For example, it is not possible to use it for outcome evaluation in a task aimed writing an essay on a serious topic.

5.6 Ground Truth

This approach leverages some predefined standards, rules or agreement to evaluate the quality of the outcomes. For some tasks, evaluating the outcome is easier than doing the task, because there are some rules that can easily determine the quality of the outcome. For example, suppose that we have a graph and want to find a path from node A to node B. Based on the structure of the graph, finding the path may be a hard task, but testing if a submitted result is indeed a correct path from A to B is easy. FoldIt uses such an approach for evaluating

outcomes [16]. The other form of ground truth is when the requester has a limited list of questions which have known answers. In this case, the requester sends some of these known-answer questions amongst the normal questions and checks to see if the worker is answering the questions honestly or not. If one of these questions is answered incorrectly, all the outcomes received from that worker may be disqualified and rejected. CrowdWeaver proposes this model for outcome evaluation. The ground truth approach is applicable only to the tasks that have identifiable ground truth standards.

5.7 Majority

This approach attempts to harness the wisdom of the crowd for evaluating the outcomes. Majority model assesses the provided outcomes based on the majority of the crowd opinions on them. Majority model is applied in two ways: (i) *redundancy* and (ii) *multilevel review*.

In the redundancy model, several instances of one task are submitted to the crowdsourcing platform and the results are collected. The answer on which the majority of workers agree will be chosen as the most appropriate answer. It is evident that this model is only applicable to crowdsourcing tasks for which the results have predefined comparable formats, e.g., the tasks for which the answer is a number or is a choice out of a specific list of options. Turkomatic employ such model of quality control.

The multilevel review is used for the tasks to which the redundancy model does not apply. For example, suppose that a requester asks for a phrase describing an image; then it is not possible to evaluate collected results using redundancy model. In multilevel review model, to evaluate the results of a crowdsourcing task, the collected results, along with the problem, are submitted to the crowdsourcing platform again as a new task (evaluation task) and the crowd is asked to check the adequacy of the results. Then the results of the evaluation tasks are collected and, based on the opinion of the majority of the workers contributing to the evaluation tasks, the outcomes provided for the main tasks are selected. TurKit [13] employs this model for outcome evaluation. The main limitation of the multilevel review is that submitting new tasks for evaluation increases the total cost of the crowdsourcing task.

5.8 Contributor Evaluation

In this model, the outcomes are evaluated based on the quality factors related to the workers who have provided outcomes. If the contributing worker has a minimum level of quality, e.g. reputation, credentials or experience, her outcomes will be accepted. Wikipedia employs this model for outcome evaluation. The articles generated by the administrators are directly applied to the system and no further evaluations are done on them.

6 Discussion and Open Issues

Dimension	**Approach**	**Leveraged Factors**	**Sample Applications**
Task	Effective Task definition	Definition User Interface Compensation Policy	[15, 11]
	Workflow Management	Granularity Definition User Interface Compensation Policy	CrowdForge CrowdWeaver Turkomatic
Worker	Open-to-All	None	Wikipedia Threadless ESP Game
	Reputation-Based	Reputation	MTurk WikiTrust StackOverflow
	Credential-Based	Credentials Experience	Wikipedia Access Control Systems
Outcome	Expert Review	Content Assessment Criterion	Innocentive
	Forced Agreement	Content	ESP Game Tag-A-Tune
	Ground Truth	Content Assessment Criterion	FoldIt
	Majority	Content Assessment Criterion	TurKit Turkomatic
	Contributor Evaluation	Assessment Criterion Reputation Credentials Experience	Wikipedia

Table 6.1: Summary of Quality Control Approaches

Crowdsourcing System	Quality Control Approaches		
	Task Management	**Worker Selection**	**Outcome Evaluation**
CrowdWeaver	Workflow Management Effective Task Definition	Reputation Credential	Ground Truth Majority (Multilevel Review)
ESP Game	Effective Task Definition	Open-to-All	Forced Agreement (Output Agreement)
InnoCentive	None	Open-to-All	Expert Review
MTurk	None	Reputation Credential	None
PeopleCloud	None	Reputation Credential	Contributor Evaluation
StackOverflow	None	Open-to-All	Expert (requester) Review
Threadless	None	Open-to-All	Majority (Multilevel Review)
TurKit	Effective Task Definition	Reputation Credential	Majority (Multilevel Review) Forced Agreement (Output Agreement)
Turkomatic	Workflow Management	Reputation Credential	Majority (Redundancy)
Wikipedia	None	Open-to-All	Expert Review

Table 6.2: Quality control approaches used in selected crowdsourcing systems

6.1 Summary

To better understand the existing quality control methods, we have summarized the quality control approaches used in crowdsourcing systems along with the quality factors they employ and have illustrated them in Table 6.1. We also study ten major crowdsourcing systems and platforms in different areas of application. CrowdWeaver [2] and Turkomatic [1] are toolkits for managing complex workflows in crowdsourcing systems. MTurk is a general-purpose online marketplace; InnoCentive[1] is a web site for crowdsourcing Research and Development (R&D) challenges which people or enterprises are facing; PeopleCloud [19] is a toolkit designed for enterprise crowdsourcing; Stackoverflow is a question and answer web site; Threadless is a T-Shirt design web site; TurKit [13]is an iterative toolkit designed for programming tasks on MTurk and Wikipedia is an open multilingual encyclopedia. The quality control approaches used in studied crowdsourcing systems are illustrated on Table 6.2.

6.2 Open Issues

Comprehensive Quality Control Framework

Various parameters contribute to the quality of a crowdsourcing task, such as user requirements, expertise and experience of the workers, time, costs, etc.; this is due to subjectivity of the quality in crowdsourcing systems. A number of quality control approaches are proposed to be used in such tasks; however, none of them is generic enough to cover all quality related requirements. In fact, it looks almost impossible to define a quality control approach which would be applicable to all kinds of tasks in every context. Thus, obtaining adequate quality control approaches for crowdsourcing tasks will necessarily always be "work in progress", which combines reusing or customizing existing approaches with custom building new ones.

Existing quality control approaches are largely hard-coded in their host systems and are not customizable based on the requirements of the requesters. There are few tools like TurKit which enable the requesters to define some basic quality control methods, but the problem with these tools is that users need to have some specific expertise, such as programming skills, for example Java for TurKit, in order to be able to use them. Most of the requesters do not have such advanced technical skills, so they cannot use such tools for quality control. It is essential to have a framework which enables requesters to easily reuse, define or customize quality control approaches without knowing how to code, how to setup servers, how to use APIs and so on. Designing such a framework is one of the challenges that require additional research in the area.

Decision making support for crowdsourcing quality control

The other important open issue regarding the quality control in crowdsourcing processes is finding a suitable quality control approach for a particular task. Crowdsourcing tasks are used in different contexts, such as evaluating items, building artifacts or playing games [5, 16]. Finding appropriate quality control approaches for a task in a particular context can be crucial for the quality of

[1] www.innocentive.com

the outcomes and also greatly impact the time and costs spent on the task. In existing systems lack of such a recommender system is obvious. Designing a recommender system which enables requesters to find quality control approaches adequate for their tasks is an important challenge worth further investigation.

Performance Verification of Quality Control Approaches

Crowdsourcing is rapidly emerging as a general model of problem solving employed to different areas of applications. Currently, most of these applications are systems which are not mission critical. Mission critical systems by their nature, deal with large amount of valuable resources. Using an approach in a critical mission system must ensure that the approach always has a certain prescribed minimal level of accuracy and timeliness. The performance of existing quality control approaches are not verified, neither formally nor empirically. Crowdsourcing systems are people centric, so quality control needs to consider several human-related parameters, such as incentives and their effect on quality, collusion and teamwork, unfair behavior and many more parameters in addition to routine factors such as cost, time and conformance to the requester's requirements. Formal or empirical verification of quality control approaches in the presence of such wide range of parameters is an important challenge for further research.

Bibliography

[1] B. H. Anand Kulkarni, Matthew Can. Collaboratively crowdsourcing workflows with turkomatic. In *The 2012 ACM Conference on Computer Supported Cooperative Work, CSCW'12*, 2012.

[2] P. A. R. E. K. Aniket Kittur, Susheel Khamkar. Crowdweaver: Visually managing complex crowd work. In *The 2012 ACM Conference on Computer Supported Cooperative Work, CSCW'12*, 2012.

[3] P. Crosby. *Quality is Free*. McGraw-Hill, NEW YORK, USA, 1979.

[4] L. De Alfaro, A. Kulshreshtha, I. Pye, and B. T. Adler. Reputation systems for open collaboration. *Commun. ACM*, 54:81–87, August 2011.

[5] A. Doan, R. Ramakrishnan, and A. Y. Halevy. Crowdsourcing systems on the world-wide web. *Commun. ACM*, 54:86–96, April 2011.

[6] K. Hoffman, D. Zage, and C. Nita-Rotaru. A survey of attack and defense techniques for reputation systems. *ACM Comput. Surv.*, 42:1:1–1:31, December 2009.

[7] J. Howe. The rise of crowdsourcing. *Wired*, June 2006.

[8] A. Ignjatovic, N. Foo, and C. T. Lee. An analytic approach to reputation ranking of participants in online transactions. In *Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Volume 01*, pages 587–590, Washington, DC, USA, 2008. IEEE Computer Society.

[9] N. J. M. Jenny J. Chen and A. D. Bradley. Opportunities for crowdsourcing research on amazon mechanical turk. In *Proceeding of The CHI 2011 Workshop on Crowdsourcing and Human Computation*, May 2011.

[10] B. John, A. Chua, and D. H.-L. Goh. What makes a high-quality user-generated answer? *Internet Computing, IEEE*, 15(1):66 –71, jan.-feb. 2011.

[11] A. Kittur, E. H. Chi, and B. Suh. Crowdsourcing user studies with mechanical turk. In *Proceedings of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, CHI '08, pages 453–456, New York, NY, USA, 2008. ACM.

[12] A. Kittur, B. Smus, and R. Kraut. Crowdforge: crowdsourcing complex work. In *Proceedings of the 2011 annual conference extended abstracts on Human factors in computing systems*, CHI EA '11, pages 1801–1806, New York, NY, USA, 2011. ACM.

[13] G. Little, L. B. Chilton, M. Goldman, and R. C. Miller. Turkit: tools for iterative tasks on mechanical turk. In *Proceedings of the ACM SIGKDD Workshop on Human Computation*, HCOMP '09, pages 29–30, New York, NY, USA, 2009. ACM.

[14] G. Little, L. B. Chilton, M. Goldman, and R. C. Miller. Exploring iterative and parallel human computation processes. In *Proceedings of the ACM SIGKDD Workshop on Human Computation*, HCOMP '10, pages 68–76, New York, NY, USA, 2010. ACM.

[15] W. Mason and D. J. Watts. Financial incentives and the "performance of crowds". *SIGKDD Explor. Newsl.*, 11:100–108, May 2010.

[16] A. J. Quinn and B. B. Bederson. Human computation: a survey and taxonomy of a growing field. In *Proceedings of the 2011 annual conference on Human factors in computing systems*, CHI '11, pages 1403–1412, New York, NY, USA, 2011. ACM.

[17] S. R. K. B. H. Steven P Dow, Anand Kulkarni. Shepherding the crowd yields better work. In *The 2012 ACM Conference on Computer Supported Cooperative Work, CSCW'12*, 2012.

[18] L. von Ahn and L. Dabbish. Designing games with a purpose. *Commun. ACM*, 51:58–67, August 2008.

[19] M. Vukovic, M. Lopez, and J. Laredo. Peoplecloud for the globally integrated enterprise. In *Service-Oriented Computing. ICSOC/ServiceWave 2009 Workshops*, volume 6275 of *Lecture Notes in Computer Science*, pages 109–114. Springer Berlin / Heidelberg, 2010.