Crowd-sourcing Literal Review

Duong Chi Thang
LSIR, I&C, EPFL

***Abstract*—Our user feedback framework requires some robust techniques in order to tackle the scalability issue of schema matching network. One approach is employing crowdsourcing/human computation models. Crowdsourcing is one of cutting-edge research areas which involves human computers to perform pre-defined tasks. In this literal review, we try to explore some certain concepts such as task, work-flow, feedback aggregation, quality control and reward system. We show that there are a lot of aspects which can be integrated into our user feedback framework.**

I. INTRODUCTION

Since 2005 [34], the phrase 'human computation' has become popular with many other similar terms, such as 'crowdsourcing', 'social computing', 'collective intelligence', etc. The term 'crowdsourcing' is used frequently in the public industry, concerning a distributed problem-solving process that involves outsourcing tasks to a network of people [1]. Whereas, the term 'human computation' is viewed as a technique in which a computational process performs its function by employing humans to do certain steps [2].

Table I summarizes common terms used in human computation.

In this review, we try to collect concepts and methods to deploy our user feedback framework as crowdsourcing tasks. For this reason, we will focus on common problems such as how to ask a question, how to design a task and how to aggregate feedbacks/answers given by users. Furthermore, we also take into account the quality of feedbacks through quality assurance/assessment methods and reward system to motive users to give correct feedbacks. Finally, we will record some information about crowdsourcing services and important statistics.

II. TASK

In crowdsourcing, task can be viewed as a problem-solving process where users/workers perform to solve certain computational problems. The common setting is given a list of questions, workers will answer them and get a reward based on their results/feedbacks.

A. Task Classification

Crowdsourcing tasks can be categorized by type or size.

[1] http://en.wikipedia.org/wiki/Crowdsourcing
[2] http://en.wikipedia.org/wiki/Human-based_computation

1) Type-based:

- **Creation task:** is the task where workers need to generate new content such as write a description or an article [21]. In this task, requesters can set some constraints on the results such as the length of the article, in what tone the workers should use.
  - Goal: produce new high quality content and the answer may be judged subjectively by requesters.
  - Example: research, writing, translation .etc
- **Decision task:** is the task where workers need to decide on existing content such as choosing which image description is better [21]. These tasks have stricter constraints since the computer need to understand the answers provided by users.
  - Goal: get accurate answer from workers.
  - Example: classification task (tagging), ranking task, clustering task .etc

2) Size-based:

- **Micro-task:** an example is a Human Intelligence Task on Amazon Mechanical Turk where completion time is short and these tasks are simple. A micro-task is considered atomic task [18].
- **Macro-task:** a complicated task that can be decomposed into smaller tasks. Macro-task can be decomposed using the task design pattern.

B. Task Combination

Computation problems can be divided into a collection of tasks where one or many workers can participate. For this reason, we need to combine the smaller tasks into a consistent process.

1) Iterative: Combines a sequence of tasks where the result of a task is the input of the next task [21]. The specialty of this type of task is that the later workers can see the results of the previous workers, which may affect the quality of the final result. Iterative task combination is most suitable for creation tasks.

- **Pros:** Increase average quality of responses.
- **Cons:** Decrease variance of responses. Negative effect on later workers since they may rely on answers from previous workers.

2) Parallel: Consists of a set of tasks that execute in parallel [21]. The specialty of this task is no worker can see the others' work.

- **Pros:** increase variance of responses.
- **Cons:** quality is the same as using only one best worker.

Term	Full Name	Description
AMT	Amazon Mechanical Turk	Amazon Mechanical Turk is an online crowdsourcing platform that connect requesters (ones who have works to be done) and workers (ones who work on tasks for money)
HIT	Human Intelligence Task	A basic and simple task that requires small time to get done. It represents a single, self-contained task [17].
GWAP	Game with a Purpose	A class of games where people perform tasks computers are unable to perform. For example, the ESP Game [36], Peekaboom [37].
Assignment	Assignment	A task can be done numerous times by different workers. Each time a task is done by a worker is called an assignment.
HIT Type	Human Intelligence Task Type	A set of HITs that have the same design template, guideline and answer set belong to a HIT type. A HIT type is also called a task group on AMT.

TABLE I
CROWD-SOURCING TERMINOLOGY

3) Hybrid: A combination of two approaches may provide a best-of-both-world solution. Workers can work on multiple iterative processes simultaneously and choose the best result to continue on. Using this hybrid approach, later workers still be affected by previous workers but since we take only the best result, the negative effect is less than that of a fully iterative process.

III. WORK-FLOW AND TASK DESIGN PATTERN

To achieve better results, we need the art of dividing a large task into a collection of smaller tasks that are combined into a so-called work-flow in which workers can check or improve each other's result. The over-heading costs and overlapping results can be avoided or reduced through selecting a suitable task design pattern/work-flow.

A. Fixed Work-flow

In this type of work-flow, the steps to execute a task are predefined before. There are two task design pattern that has been studied in the literature:

- **Find-fix-verify:** see section Defensive Task Design in [2]. This design pattern is effective in countering lazy or overeager workers.
- **Solve-decompose-combine:** in [18], the authors propose an approach to harness the workers on AMT to design the task workflow for a task. The requesters can monitor the decompose process to intervene if required.
  - Solve: The requesters ask the worker to solve a task.
  - Decompose: Ask a worker to decompose a task into various subtasks.
  - Combine: Ask the worker to merge the results from the subtasks.
  - Verify: these steps may also include the verification at the end of each step to improve the quality of each step's result.

B. Dynamic Work-flow

In this type of work-flow, the steps may not be predefined. The sequence to execute the tasks and which task to post to workers are decided automatically or semi-automatically.

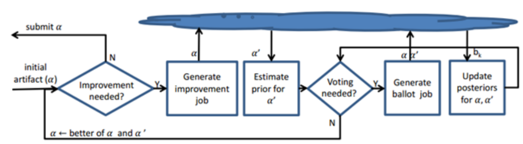

Fig. 1. Dynamic Work-flow

- **Semi-automatic:** tasks are designed with the help of workers. Two prominent examples are depicted in [41] and [18]. In [41], a travelling plan is designed using the workers on AMT. The requesters define several constraints that for their plan. First, workers will pose ideas that are essentially tasks for other workers to complete if they believe the idea is reasonable. The system also help by identifying violations: if the travelling plan does not satisfy a constraint, a new task will be asked to resolve this violation.
- **Automatic:** tasks are designed automatically based on the quality of the received results [38] or expecting goals [30]. In [38], the next question to ask or how many workers need to ask are defined automatically based on decision theory. The Fig. 1 displays the framework that the authors proposed. Basically, the system harnesses the crowd to answer two questions: if the current work need improvement and how to resolve improvement disagreement between workers through voting.

C. Quality-oriented task design guideline

1) The more time workers spend on task, the better quality the result:

- Complex task design: Higher task complexity drastically discourages malicious workers from attempting to cheat [9].
- Variance of task: Greater variability and more context changes discourage malicious workers as the task appears less susceptible to automation or cheating, in other words less profitable [9].

- Self-report: Add self-report response from workers since an inconsistent self-report response could indicate a malicious worker.

2) The more similar to an expert works, the better quality the result: In order to improve quality, designing task should mimic the way an expert solve the problem in the task. In [17], the authors argue the need to add verifiable questions to a task by experimenting between the quality of no-verifiable questions and verifiable questions. Since workers need to concentrate to answer these questions and requesters can also check if these answers are correct, workers refrain from cheat.

- Task must have explicitly verifiable questions as part of the task.
- Task must be designed so that giving spam or malicious answers must take approximately equal time to giving correct answers.
- Use multiple ways to detect suspect responses. For example, short task duration may indicate a malicious worker. Moreover, self-report responses such as the difficulty of the task, the interestingness can also be used to find out inconsistent responses.

3) Tasks are designed so that workers can work on the tasks effectively: The authors of [41], [25] - after developing a collaborative platform for workers to solve a complicated problem - has come up with the following guidelines:

- Keep the crowd, the solution and the context together: through having a single interface, workers are easily to coordinate and communicate. On AMT, this is achieved via HIT template many HITs have the same user interface.
- Interactions are less controlled but still structured: workers can freely contribute to the solving process but they are still constraint by the constraints. This guideline is suitable for creation tasks.
- A fluid way to refine goals: requesters can change their goals as workers still working on the previous goals.
- The task design can follow the item-response theory which is the science of inferences and scoring for tests, questionnaires and examinations [29]. Item-response theory can be used to design the template and also the ordering of HITs.

IV. Question

Questions take a significant role to introduce workers to perform crowdsourcing tasks. There are many aspects should be considered such as complexity, psychology, education, etc. There are still a lot of interesting research problems to take into account.

A. Question Classification

In general, questions can be classified by their kind of answers.

- **Closed-class question:** is a question in which workers can select answers from a set of possible answers. A closed class question is used in a decision task where answers are known in advanced [9].
- **Open-class question:** is a question where workers can type their answers in free text form. An open class question is used in a creation task [9].

V. Answer - Feedback Aggregation

There are various aggregation methods [26], [27]. Two well-known ones are majority voting and expectation maximization.

Beside these two prominent methods, probabilistic models and other sophisticated approaches are built to aggregate answers and also evaluate worker quality. However, these methods depend on the assigned tasks and cannot be scale to other tasks.

A. Majority voting

From the results collected from many workers, select the ones which workers agree most. Many research have shown that majority voting is superior to other sophisticated approaches [34].

- **Pros:** Simple. Robust to errors. Best in case of similar workers.
- **Cons:** Susceptible with random guesses, mistakes or correct by chance. Subjective answers: different expertise has different answers, bias users. Discard minority votes, only keep one answer.

B. Expectation Maximization (EM)

In the EM algorithm, there are two phases that a set of workers and their responses are evaluated. The first phase is expectation where the worker's quality is calculated based on their answers and the current correct answers. In the maximization phase, the current correct answers are modified according to the newly-computed worker quality. These two phases are executed until convergence is reached [39], [38], [16]. Majority voting can be considered a special case of EM where worker quality are equal and stable.

In [38], the quality of an artifact is calculated based on the worker quality and worker quality is calculated using the estimation of previous works . In [16], the authors extend the EM algorithm by introducing soft labels to differentiate between bias and malicious workers.

- **Pros:** Evaluate worker quality dynamically
- **Cons:** [28] Requires batch post processing. If results not satisfied, must recollected and rerun. Large collection increases cost. Offline delays and permit workers submit useless works.

C. Worker Classification

Classifying workers is an important process to control the quality of feedbacks. From that, we can evaluate the feedbacks more confidently with respect to the quality of associated workers.

- **Lazy worker:** a lazy worker is a worker who gives the same answer for every closed class question [16].
- **Malicious worker:** a malicious worker is a worker who attempts to give incorrect answer.

- **Biased worker:** is a worker who is affected by cultural or outside conditions. For example, when classifying films, parents would give stricter answers than normal workers. A biased workers answers are consistently and predictably incorrect. We are able to reverse the error to find the correct one in contrast with a malicious worker whose answer cannot be reversed [16].
- **Normal worker:** is a worker whose answers can be correct or not. We can denote the probability that the answer is correct by p [16].
- **Expert:** is a normal worker with $p = 1$.

VI. Quality

Quality is an essential problem which requires a lot of managing and controlling methods. The more quality of feedbacks is, the more knowledge we can achieve by aggregating the feedback results.

A. *Quality assurance methods*

In order to ensure quality, several methods can be used. However, these methods are based on some meta-heuristics [1] defining how high quality responses are made. A high quality answer may have some of these characteristics:

- Hard to improve [38]
- Increase with user's effort [3]

Therefore, quality assurance has three aspects to be taken care of [33]:

- Ensuring worker understanding of the requested task and try to perform it well
- Cleaning up occasional errors
- Detecting and preventing cheat

Table II summarizes the strength and weaknesses of common quality assurance methods.

1) Pre-screening Test: Pre-screening [28] workers is a simple strategy: set up multiple-choice questions and ban people who do not pass the test. For example, on Amazon Mechanical Turk, workers must pass a qualification test designed by requesters in order to work on the requester's tasks.

2) Redundancy: A task can be done by many workers and these works are combined using a feedback aggregation method such as majority voting [33]. For example, in reCAPTCHA which aims to digitize books through CAPTCHA, a word scanned from a book will displayed to users as CAPTCHA. This word will be used as CAPTCHA for various users and through majority voting, the word will be considered a correct word get from the scan.

3) Forced Agreement - Game with a Purpose: In Games with a Purpose (GWAP), there are mechanisms to check if the playersinput matches the others. If their inputs match, winning condition is met. Therefore, in order to win these games, players must be truthful to each other $\Longrightarrow$ correct answers. There are two kinds of agreements:

- **Input Agreement:** two players are given two inputs only known by game. They must describe the input they received and decided whether they are given the same input [19], [35]. Fig. 2 illustrates this method.

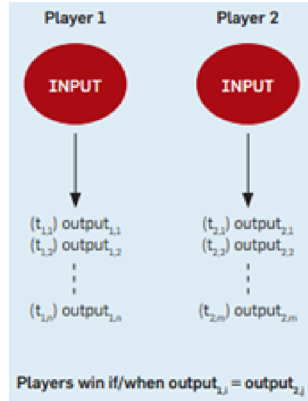

Fig. 2. Input-Agreement for quality assurance

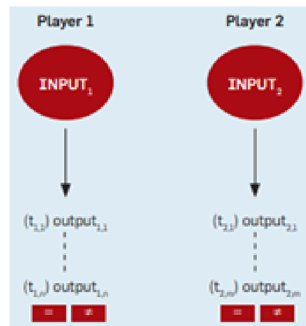

Fig. 3. Output-Agreement for quality assurance

- **Output Agreement:** two players are given the same input and describe the input they receive. If their describes are the same, the answer is accepted [35]. Fig. 3 illustrates this method.

B. *Quality assessment methods*

A quality assessment method is needed for the following reasons:

- Human makes mistake
- Human have bias or objective views
- Human may give malicious or spam answer

Table III summarizes the pros and cons of well-known quality assessment methods.

1) Constraint-based filtering: In [5], the authors define a transitivity constraint of the answers from workers. If some answers do not conform to this constraint, these answers are considered incorrect and the worker is malicious. Moreover, in [41], the authors propose a system where unsatisfied parts of the answers are posed as new tasks for workers to work on them.

2) Multi-level review: In the multilevel review method, a big and complicated task is decomposed into many consecutive steps. The output of one step is the input for the next step.

One example is the Soylent word processing [2]. The word processing task is decomposed into 3 steps: find-fix-verify: "find" step is used to identify parts of the documents need to

	Advantages	**Disadvantages**
Pre-screening test	• Simple to implement • Preventing unskilled or unethical workers from entering a task	• Affect bad and good workers alike • Require continued incentive to control quality • Vulnerable to scammers • Test may be designed manually
Redundancy	• Reduce influence of errors • Can be used to catch malicious workers or find real distribution of answers	• Increased cost • May require a reputation system to evaluate worker quality
Forced agreement	Fun which elicit workers participation	Hard to design GWAP, need a platform to operate

TABLE II
ADVANTAGES AND DISADVANTAGES OF QUALITY ASSURANCE METHODS

be improved. The ”fix” step will modify the identified parts to improve quality. Finally, the ”verify” step will check whether the fixed patches are correct.

Another example is Turkomatic [18] where the task are processed in many steps: price-divide-merge-solve and verify. In the price step, an initial pricing HIT is placed on AMT to find out the overall task goal can be achieved. If it is possible, another worker will divide the overall task into smaller task. The results of this step may be reviewed as a task and merge together. Next, these subtasks are solved and verify by a set of workers.

3) Expert review - ground-truth checking: Use some test questions - questions to which answers are known [12], [28] - to test the quality of workers or counter spam. Works done by workers may be reviewed by experts to check for relevance and accuracy [34].

4) Automatic check: Some questions are hard to find the results but the results are easy to check. For example, in the fold.it protein folding game [6], its hard to find out the optimal structure of the proteins but when the results is found, it can be checked to be correct or not.

VII. REWARD

As in many working environment involving human workers, the reward has a vast effects on motivation and performances of their contribution. There is a lot of research studies that tried to develop theory to capture the relationship between reward size and quality of results.

A. Economic models

Using money as motivation [12], [3], [22], requesters can design their payment plans in order to improve the quality of workers' production with less money. In [22], the authors study the relationship between payments and the quality of the results. The study shows that increasing payment may not improve performance of workers but the quantity of workers participated in the tasks. In [12], the authors use Pareto optimality which is an economic model to compute the optimal payments for worker based on their skills and the quality of their works.

B. Payment Guideline

1) Quota-based scheme is more effective than piece-based scheme: In [22], the authors run an experiments on a word puzzle game where payments are based on the number of words found (piece-based) or the number of puzzle successfully completed (quota-based). They found that quota-based works better than piece-based.

2) The higher quality of the result, the higher salary: [12]

- Directly relate payment to the worker ability: since it increases the worker's payment, this will affect the workers' attempt to cheat.
- Decrease worker ability substantially for incorrect answers: this servers as a penalty for the workers who attempt to cheat since it would take them longer to regain that ability.
- Increase worker quality slowly for correct answers: since their ability evaluation decrease significantly when cheating but decrease slowly when being diligent, workers may afraid to cheat.
- Require worker to maintain a minimum reserve of quality score.

There are some other theories that research on the payment of workers when receiving a task such as decision theory and game theory.

C. Reputation System

Beside payment, the quality of workers' works can affect their reputation. In order to manage worker reputation, one may need to build a reputation system [3]. A reputation system keeps track of worker past actions to evaluate their current reputation. Payment can also be based on worker reputation. In contrast with the economic models, a reputation system not only decides payments based on current works but also from the past. For example, Amazon Mechanical Turk (AMT) currently monitors worker responses. If a worker keeps giving incorrect answer, he/she is prohibited to access future tasks.

VIII. CROWDSOURCING SERVICES

Crowdsourcing has become a interesting business in online industry recently. Many platforms were introduced, such as well-known Amazon Mechanical Turk, CrowdFlower, Taskcn,

	Advantages	**Disadvantages**
Constraint based filtering	Easy and efficient for malicious or correct answer	Few tasks have constraints on output
Multilevel review	Complexity in each step decreases	More time to complete, increased cost
Groundtruth checking	• Simple and efficient and adjustable passing grades • Explicitly measuring worker accuracy • Transparent to workers, workers can learn from mistakes	• Need to compute ground-truth • hurt completion time, hard to find ground truth for subjective task
Automatic check	Interactive since workers can receive feedbacks immediately	• Few problems can be checked automatically • Require an independent component to check for correctness

TABLE III
ADVANTAGES AND DISADVANTAGES OF QUALITY ASSESSMENT METHODS

etc. Firstly, we introduce some features commonly provided by crowdsourcing platforms. Later we focus on the programming aspects to implement our user feedback framework through web services and APIs.

A. Amazon Mechanical Turk (AMT)

Amazon Mechanical Turk is an online crowdsourcing platform that connect requesters (ones who have works to be done) and workers (ones who work on tasks for money). The requesters' works are decomposed into small, simple tasks called HITs that workers select to complete. Requesters need to break down work into tasks and control quality of the results by themselves. Most of these tasks are easy for human to work on but difficult for computers to solve.

1) Quality management mechanisms on AMT: There are three quality management methods on AMT [4], [17]:

- A work can be rejected if it has low quality. A worker can also be rejected to participate in some HITs or even blocked.
- A job can be done by different workers.
- Worker must meet some qualifications defined by requester to work on a HIT.

AMT lacks the ability to incorporate golden answers to evaluate worker quality.

2) Steps to post tasks to AMT:

Step 1. Define goals and key components of the project.
Step 2. Break the project into various tasks.Design the HIT template that will be used through the tasks.
Step 3. Publish HITs to AMT. A HIT can have multiple assignments (worked by many workers).
Step 4. Pre-screening test: designing qualification test that workers need to pass to work on tasks.
Step 5. Review the works done by workers to approve or reject their works.

B. CrowdFlower

Unlike AMT, CrowdFlower is a service that bridges the need between requesters and AMT. It handles work decomposition, task workflow and result quality for requesters. Tasks decomposed by CrowdFlower can be uploaded to AMT for workers to work on. One of the quality assessment method that CrowdFlower provides but AMT lacks of is ground truth value checking.

CrowdFlower was used for conducting a crisis relief experiment in [13]. The most significant contribution of CrowdFlower to this experiment is the scalability of the pool of crowd resources (workers, machines, ...).

C. Taskcn

Taskcn (Taskcn.com) is viewed as a Witkey site - a type of knowledge sharing market where questions are posed by requesters and answered by other users. User whose answer is correct gets a monetary reward. Taskcn differs from AMT that taskcn focuses on creation tasks where the majority of tasks on AMT are decision tasks. For example, a company may pose a logo design contest to taskcn. Solvers need to research about the company in order to design a logo that suitable for the company culture.

Taskcn was used for examining the behavior of users in [40]. Taskcn has 1.7 million registered users, with around 3100 tasks and 543000 answers (up to 2008).

IX. STATISTICS AND APPLICATIONS

In this section, we capture some important statistics and indicators to provide an overview about the growth of crowdsourcing era in both industry and research.

A. Amazon Mechanical Turk

Although workers from any country can work on tasks on AMT, only Indian and US workers can receive money directly to their bank accounts. Therefore, the majority of workers on AMT are Indian and US citizens as described in Figure 4 [32].

Moreover, the workers are young which more than 50% of AMT workers have age below 34. In addition, the workers keep getting younger as the average age decreases through time as illustrated in Figure 5 [32].

An important finding is that the workers on AMT are highly educated as described in Table IV [32]. In an older survey by Ipeirotis, we get a consistent result that most AMT workers have a bachelor degree or higher as showed in Figure 6 [15].

	Nov 08	May 09	Aug 09	Nov 09
US	• 32% Bachelors • 11% Graduate	• 34% Bachelors • 14% Graduate	• 34% Bachelors • 19% Graduate	• 38% Bachelors • 17% Graduate
India	• 69% Bachelors • 29% Graduate	• 56% Bachelors • 18% Graduate	• 56% Bachelors • 13% Graduate	• 45% Bachelors • 21% Graduate

TABLE IV
EDUCATION OF AMT WORKERS

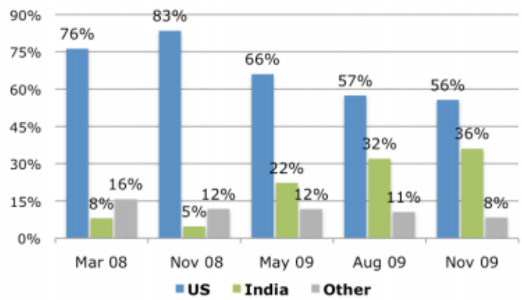

Fig. 4. Nationality distribution of AMT workers

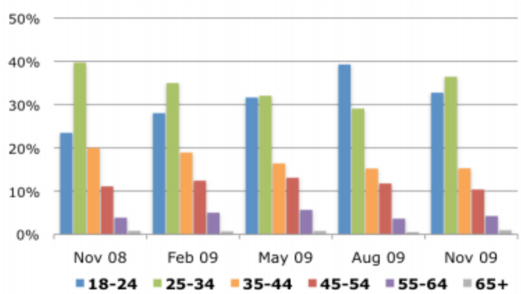

Fig. 5. Age distribution of AMT workers

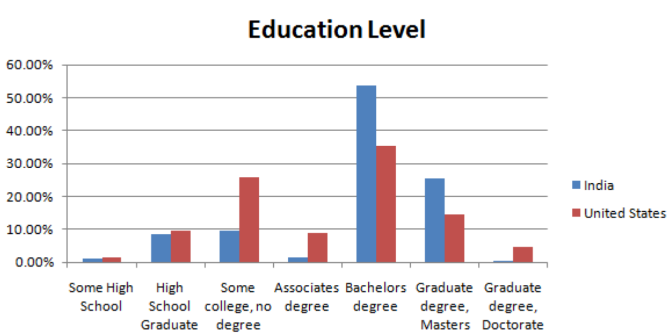

Fig. 6. Education of AMT workers

B. Applications

There are a lot of surveys and tutorials [7], [8], [20], [31], [14], [24] toward fostering more discussions on applying crowdsourcing to various research areas, including data integration [10], [23], [11], world-wide-web, and information retrieval. Especially in data management perspective, many computationally difficult tasks are addressed such as entity resolution, schema matching, object recognition, outlier detection, etc. In their surveys, the authors raised a wide-range of issues varying from task scheduling, cost optimization to privacy and social issues. However, there is still no concrete direction that has been proposed for schema matching network.

REFERENCES

[1] Duong Tuan Anh, Vo Hoang Tam, and Nguyen Quoc Viet Hung, *Generating complete university course timetables by using local search methods.*, RIVF, 2006, pp. 67–74.

[2] Michael S. Bernstein, Greg Little, Robert C. Miller, Björn Hartmann, Mark S. Ackerman, David R. Karger, David Crowell, and Katrina Panovich, *Soylent: a word processor with a crowd inside*, Proceedings of the 23nd annual ACM symposium on User interface software and technology (New York, NY, USA), UIST '10, ACM, 2010, pp. 313–322.

[3] Trevor Burnham and Rahul Sami, *A reputation system for selling human computation*, Proceedings of the ACM SIGKDD Workshop on Human Computation (New York, NY, USA), HCOMP '09, ACM, 2009, pp. 54–57.

[4] Chris Callison-Burch and Mark Dredze, *Creating speech and language data with amazon's mechanical turk*, Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk (Stroudsburg, PA, USA), CSLDAMT '10, Association for Computational Linguistics, 2010, pp. 1–12.

[5] Kuan-Ta Chen, Chen-Chi Wu, Yu-Chun Chang, and Chin-Laung Lei, *A crowdsourceable qoe evaluation framework for multimedia content*, Proceedings of the 17th ACM international conference on Multimedia (New York, NY, USA), MM '09, ACM, 2009, pp. 491–500.

[6] Seth Cooper, Firas Khatib, Adrien Treuille, Janos Barbero, Jeehyung Lee, Michael Beenen, Andrew Leaver-Fay, David Baker, Zoran Popovic, and Foldit Players, *Predicting protein structures with a multiplayer online game.*, Nature **466** (2010), no. 7307, 756–760.

[7] AnHai Doan, Michael J. Franklin, Donald Kossmann, and Tim Kraska, *Crowdsourcing applications and platforms: A data management perspective*, PVLDB **4** (2011), no. 12, 1508–1509.

[8] Anhai Doan, Raghu Ramakrishnan, and Alon Y. Halevy, *Crowdsourcing systems on the world-wide web*, Commun. ACM **54** (2011), no. 4, 86–96.

[9] Carsten Eickhoff and Arjen P. de Vries, *How crowdsourcable is your task?*, Workshop on Crowdsourcing for Search and Data Mining (CSDM) (Hong Kong, China), 2011.

[10] Avigdor Gal, Michael Katz, Tomer Sagi, Matthias Weidlich, Karl Aberer, Hung Quoc Viet Nguyen, Zoltán Miklós, Eliezer Levy, and Victor Shafran, *Completeness and ambiguity of schema cover*, CoopIS, 2013, pp. 241–258.

[11] Avigdor Gal, Tomer Sagi, Matthias Weidlich, Eliezer Levy, Victor Shafran, Zoltán Miklós, and Nguyen Quoc Viet Hung, *Making sense of top-k matchings: A unified match graph for schema matching*, 2012, p. 6.

[12] Craig Gentry, Zulfikar Ramzan, and Stuart Stubblebine, *Security protocols*, Springer-Verlag, Berlin, Heidelberg, 2009, pp. 177–180.

[13] Vaughn Hester, Aaron Shaw, and Lukas Biewald, *Scalable crisis relief: Crowdsourced sms translation and categorization with mission 4636*, Proceedings of the First ACM Symposium on Computing for Development (New York, NY, USA), ACM DEV '10, ACM, 2010, pp. 15:1–15:7.

[14] Nguyen Quoc Viet Hung, Saket Sathe, Duong Chi Thang, and Karl Aberer, *Towards enabling probabilistic databases for participatory sensing*, CollaborateCom, 2014, pp. 114–123.
[15] P. Ipeirotis, *Mechanical turk: The demographics*, Computer Scientist in a Business School, 2008.
[16] Panagiotis G. Ipeirotis, Foster Provost, and Jing Wang, *Quality management on amazon mechanical turk*, Proceedings of the ACM SIGKDD Workshop on Human Computation (New York, NY, USA), HCOMP '10, ACM, 2010, pp. 64–67.
[17] Aniket Kittur, Ed H. Chi, and Bongwon Suh, *Crowdsourcing user studies with mechanical turk*, Proceedings of the twenty-sixth annual SIGCHI conference on Human factors in computing systems (New York, NY, USA), CHI '08, ACM, 2008, pp. 453–456.
[18] Anand Kulkarni, Matthew Can, and Hartmann, *Collaboratively crowdsourcing workflows with turkomatic*, Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work (New York, NY, USA), CSCW '12, ACM, 2012, pp. 1003–1012.
[19] Edith Law and Luis von Ahn, *Input-agreement: a new mechanism for collecting data using human computation games*, Proceedings of the 27th international conference on Human factors in computing systems (New York, NY, USA), CHI '09, ACM, 2009, pp. 1197–1206.
[20] Matthew Lease and Emine Yilmaz, *Crowdsourcing for information retrieval*, SIGIR Forum **45** (2012), no. 2, 66–75.
[21] Greg Little, Lydia B. Chilton, Max Goldman, and Robert C. Miller, *Exploring iterative and parallel human computation processes*, Proceedings of the ACM SIGKDD Workshop on Human Computation (New York, NY, USA), HCOMP '10, ACM, 2010, pp. 68–76.
[22] Winter Mason and Duncan J. Watts, *Financial incentives and the "performance of crowds"*, Proceedings of the ACM SIGKDD Workshop on Human Computation (New York, NY, USA), HCOMP '09, ACM, 2009, pp. 77–85.
[23] Hung Quoc Viet Nguyen, Tri Kurniawan Wijaya, Zoltán Miklós, Karl Aberer, Eliezer Levy, Victor Shafran, Avigdor Gal, and Matthias Weidlich, *Minimizing human effort in reconciling match networks*, ER, 2013, pp. 212–226.
[24] Quoc Viet Hung Nguyen, Son Thanh Do, Tam Nguyen Thanh, and Karl Aberer, *Privacy-preserving schema reuse*, DASFAA, 2014, pp. 234–250.
[25] Quoc Viet Hung Nguyen, XuanHoai Luong, Zoltan Miklos, ThoThanh Quan, and Karl Aberer, *Collaborative schema matching reconciliation*, CoopIS, 2013, pp. 222–240.
[26] Quoc Viet Hung Nguyen, Thanh Tam Nguyen, Ngoc Tran Lam, and Karl Aberer, *Batc: a benchmark for aggregation techniques in crowdsourcing*, SIGIR, 2013, pp. 1079–1080.
[27] Quoc Viet Hung Nguyen, Tam Nguyen Thanh, Tran Lam Ngoc, and Karl Aberer, *An evaluation of aggregation techniques in crowdsourcing*, WISE, 2013, pp. 1–15.
[28] David Oleson, Alexander Sorokin, Greg P. Laughlin, Vaughn Hester, John Le, and Lukas Biewald, *Programmatic gold: Targeted and scalable quality assurance in crowdsourcing*, Human Computation, 2011.
[29] Aditya Parameswaran and Neoklis Polyzotis, *Answering queries using humans, algorithms and databases*, Conference on Inovative Data Systems Research (CIDR 2011), Stanford InfoLab, January 2011.
[30] Aditya Parameswaran, Anish Das Sarma, Hector Garcia-Molina, Neoklis Polyzotis, and Jennifer Widom, *Human-assisted graph search: it's okay to ask questions*, Proc. VLDB Endow. **4** (2011), no. 5, 267–278.
[31] Alexander J. Quinn and Benjamin B. Bederson, *Human computation: a survey and taxonomy of a growing field*, Proceedings of the 2011 annual conference on Human factors in computing systems (New York, NY, USA), CHI '11, ACM, 2011, pp. 1403–1412.
[32] J. Ross, L. Irani, M. Silberman, A. Zaldivar, and B. Tomlinson, *Who are the crowdworkers?: shifting demographics in mechanical turk*, Proceedings of the 28th of the international conference extended abstracts on Human factors in computing systems, ACM, 2010, pp. 2863–2872.
[33] A. Sorokin and D. Forsyth, *Utility data annotation with amazon mechanical turk*, Computer Vision and Pattern Recognition Workshops, 2008. CVPRW '08. IEEE Computer Society Conference on, june 2008, pp. 1 –8.
[34] L. von Ahn, *Human computation*, Design Automation Conference, july 2009, pp. 418 –419.
[35] L. von Ahn and L. Dabbish, *Designing games with a purpose*, Commun. ACM **51** (2008), no. 8, 58–67.
[36] Luis von Ahn and Laura Dabbish, *Labeling images with a computer game*, Proceedings of the SIGCHI conference on Human factors in computing systems (New York, NY, USA), CHI '04, ACM, 2004, pp. 319–326.
[37] Luis von Ahn, Ruoran Liu, and Manuel Blum, *Peekaboom: a game for locating objects in images*, Proceedings of the SIGCHI conference on Human Factors in computing systems (New York, NY, USA), CHI '06, ACM, 2006, pp. 55–64.
[38] with Dai Peng Weld, Daniel S and Mausam, *Decision-theoretic control of crowd-sourced workflows*, Artificial Intelligence (2010).
[39] P. Welinder and P. Perona, *Online crowdsourcing: Rating annotators and obtaining cost-effective labels*, Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on, june 2010, pp. 25 –32.
[40] Jiang Yang, Lada A. Adamic, and Mark S. Ackerman, *Crowdsourcing and knowledge sharing: strategic user behavior on taskcn*, Proceedings of the 9th ACM conference on Electronic commerce (New York, NY, USA), EC '08, ACM, 2008, pp. 246–255.
[41] Haoqi Zhang, Edith Law, Robert C. Miller, Krzysztof Z. Gajos, David C. Parkes, and Eric Horvitz, *Human computation tasks with global constraints*, Proceedings of CHI'12, 2012, To appear.