Knowledge and Information Systems (2019) 60:1–24
https://doi.org/10.1007/s10115-018-1235-5

SURVEY PAPER

Overview of the crowdsourcing process

Lobna Nassar[1] · **Fakhri Karray**[1]

Received: 9 October 2016 / Revised: 17 March 2018 / Accepted: 14 April 2018 /
Published online: 13 July 2018

Abstract
A decade ago, the crowdsourcing term was first coined and used to represent a method for expressing the wisdom of the crowd in accomplishing two types of tasks. One type includes tasks that need human intelligence rather than machines, and the other type covers those tasks that can be accomplished with a higher time and cost efficiency using the crowd rather than employing experts. The crowdsourcing process contains five modules: The first is designing *incentives* to mobilize the crowd to do the required task. This step is followed by four modules for collecting and assuring *quality* and then *verifying* and *aggregating* the received information. The verification and quality control can be done for the tasks, collected data and the participants by having more participants answer the same question or accepting answers only from experts to avoid errors from unreliable participants. Methods of discovering topic *experts* are utilized to discover reliable candidates in the crowd who have relevant experience in the discussed topic. Expert discovery reduces the number of needed participants per question which reduces the overall cost. This work summarizes and reviews the methods used to accomplish each processing step. Yet, choosing a specific method remains application dependent.

Keywords Crowdsourcing · Incentives · Verification · Aggregation · Quality assurance · Quality control (QC) · Expert discovery

1 Introduction

The crowdsourcing term refers to using the wisdom of the crowd in solving a problem. It is considered as an effective and efficient way of finding information. For example, in [45] it is found that the annotation results coming from the non-expert crowd and TREC assessors have compatible quality. This makes the process of creating ground truth annotations for new datasets using crowdsourcing cheaper and faster with compatible quality. In [36], it is proved that the use of crowdsourcing with its low overall costs and fast completion rates

✉ Lobna Nassar
lnassar@uwaterloo.ca

Fakhri Karray
Karray@uwaterloo.ca

[1] Electrical and Computer Engineering, University of Waterloo, 200 University Avenue. West, Waterloo, ON N2L 3G1, Canada

performs better than using manual annotation by experts in research laboratories. In [21], the Crowdsourcing Software Engineering definition is based on Jeff Howe's definition of crowdsourcing (the most widely accepted crowdsourcing definition), and it is defined as follows:

"Crowdsourced Software Engineering is the act of undertaking any external software engineering tasks by an undefined, potentially large group of online workers in an open call format" [21].

An anatomy is introduced in [8] where a comparison between the information retrieval and the crowdsourcing processes is discussed. It is said that the main paradigm for information retrieval is a library and the main paradigm for the social search over social media is a village. In the first people find information in authorized trusted books, and in the later the information is trusted when it comes from known people in the village.

Crowdsourcing can be seen as a method of finding information through social networks which resembles looking for people to find answers instead of searching documents. Since the main source of information is people, there should be a way to encourage them to do the task by designing proper incentives.

However, not all these people are reliable; therefore, the same task or question can be asked to more than one person and a way to aggregate and verify the answers to find the best answer for each question should be found. Quality control methods can be applied when designing the tasks, collecting data and selecting the participants. Searching for the person with a high experience in the discussed topic and known by others to be capable of doing the task can enhance the reliability of found information and reduce the aggregation and verification costs. Therefore, way to discover topic experts is considered as an important step in the crowdsourcing process. These steps of the crowdsourcing process are discussed in detail in this work as well as a review of the methods utilized for achieving each step.

Section 2 describes the crowdsourcing stages including: incentives deployed to mobilize participants to contribute to the crowdsourced tasks, quality assurance approaches, collected information verification methods, ways to aggregate participants' contributions and lists topical experts' discovery along with ranking techniques. Section 3 concludes the paper.

2 The crowdsourcing process

The crowdsourcing process mainly contains five modules: It starts by designing *incentives* then *quality control* methods for users, tasks and collected data. This is followed by *collecting*, *aggregating* and *verifying* received data as in Fig. 1. The verification can be done by having more participants to answer the same question or accepting answers from experts only to avoid errors from unreliable participants. *Expert discovery* can reduce the number of needed participants per question which reduces the cost. Different methods are discussed for each stage, yet selecting the best one remains application dependent. Table 1 is used as a guide when selecting the methods that best fit the considered application.

2.1 Incentives design

Despite the improvement in the software engineering, there are some tasks that still need human intelligence and cannot be easily achieved with full automation. Human collective brainpower has enormous potential to address problems computers cannot tackle on their own.

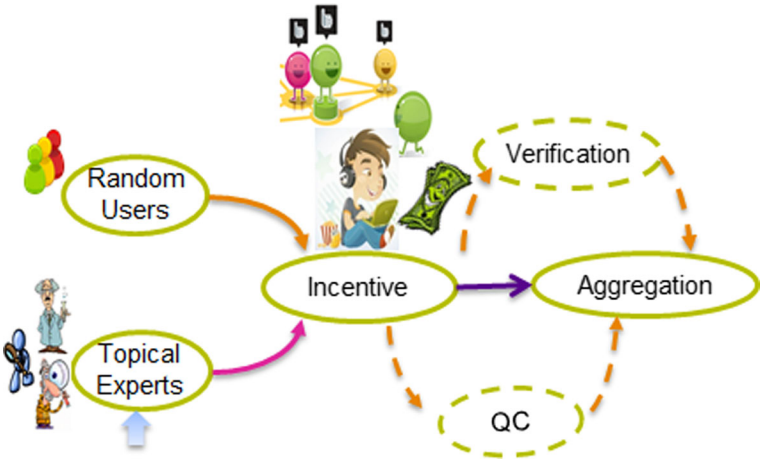

Fig. 1 Crowdsourcing process

Hence, people should be given some incentives to become part of this collaborative computation [14]. Incentives can be intrinsic (personal enthusiasm or altruism) or extrinsic (incentives or monetary reward) [33]. Both can motivate honest users to participate. It is argued that intrinsic incentives' positive effect on the outcome's quality is more significant than the extrinsic ones. Money speeds up the attraction of participants, but more money does not directly affect the quality, whereas the payment assigning method can affect the quality [33]. Insufficient incentives result in dropping out from the search experiments [29]. In this section, the methods used to mobilize the crowdsourcing contributors to rapidly participate are discussed.

- *Entertainment* Having some fun is a need for many Internet users; therefore, designing the computation tasks as paired games like in [27] is one of the successful ways to motivate a large number of people to participate for free. These fun games or what they call "games with a purpose" should be correctly and efficiently designed to achieve the objectives behind using them, but most importantly they should be enjoyable since this is the main motive behind playing them. Some of the famous games designed for this purpose, such as the *ESP* and *Peekaboom* games for labeling random images, demonstrate how humans, as they play, can solve problems that cannot yet be solved by computers. *Phetch* is another game that annotates images with descriptive paragraphs, and Verbosity collects common sense facts to train reasoning algorithms. Games have been developed and used in areas as security, computer vision, Internet accessibility, adult content filtering, language translation, monitoring of security cameras, improving Web search and text summarization [26].

On the other hand, these paired games have some drawbacks when compared to other applications that rely on crowdsourced annotations mainly by experts like the ImageNet in [16]. First, ImageNet annotates the images in a hierarchical way from the basic general semantic category to more detailed semantic sets, while the *ESP* game labels, for example, largely concentrate on the basic level of the semantic hierarchy of images and the dataset is not fully accessible to the public. Second, the centralization of the game-based approaches is criticized in [3] since it makes it expensive to have proper scale, which hinders the development of new games. Besides, to achieve reasonable scale the game becomes more complicated

Table 1 Guide for each of the 5 crowdsourcing stages: incentives design (ID), quality control (QC), verification (*V*), information aggregation (*A*) and topical experts discovery (TED)

Papers	ID	QC	*V*	*A*	TED
[3]	√	√	√	√	
[8]					√
[4, 10]	√				
[12]	√				√
[14]	√		√		
[15]				√	
[16]	√		√		
[26]	√			√	
[27]	√				
[29]	√				√
[18, 28]			√	√	
[33]	√	√	√	√	
[34]		√		√	
[35]		√			
[36]				√	√
[39]				√	
[41]			√	√	√
[42]		√	√		
[44]		√	√		
[37, 38]				√	√
[1, 5, 6, 9, 11, 19, 22, 23, 30, 31, 36, 40]		√			
[2, 7, 17, 24, 32, 43, 46, 47]					√
[13, 20, 48, 49]			√		

to design with a proper interface. Third, *ESP* players have freedom of expression and can use ambiguous labels that might have more than one meaning. In ImageNet, the annotator is given a limited set of choices to avoid ambiguity.

- *Social Recognition* Attention is another important driver for contributions [4]; e.g., the main driving force behind YouTube contributions is viewers' attention [10]. Status and recognition are contribution motivators in the open source community [4].
- *Financial Compensation* Missing financial incentives can cause individuals to be reluctant to pass the search to others even if it would cost them a single mouse click to forward the message [29]. A recursive incentive mechanism can be used as in [14] where the money values are recursively distributed among the answerer and the people along the referral path leading to the answer, i.e., all those who forwarded the question along the way to reach the answerer. On the other hand, other applications, like the Amazon's Mechanical Turk (AMT), use money incentive to encourage solving human intelligent tasks (HITs) that are more effectively and efficiently solved by humans rather than by machines. In [12], the authors need to provide monetary incentives plus other incentives for inducing contributions for the problem of efficiently obtaining accurate probability assessments about the occurrence of an event or proposition truth. They made use of routing scoring

rules to design an incentive mechanism that induces contributors to truthfully update probability assessments, and route tasks to those who can best refine predictions.

2.2 Quality control (QC)

The crowdsourcing process is expanded to include verification and quality assurance of the process. Quality control approaches for participants' selection, task design and collected data are discussed in this section.

2.2.1 Workers QC

Worker quality (accuracy and trustworthiness) affects the overall quality of the crowdsourcing process. In [3], the authors highlight the importance of protection against cheating by spammers, fixing normal workers errors, ensuring the full understanding of the requested task by workers and their will to do their best on the overall outcome quality.

In [35], it is claimed that errors coming from biased workers are recoverable. According to the authors, the resolvable systematic bias that might be caused by high-quality workers might not be recognized by naive error rate measurement methods. Hence, these methods can underestimate the true worker quality which leads to unthoughtful blocks of valid workers. In [33], it is stated that workers should be provided with a clear task description (the clarity affects the quality of the response) and eligibility criteria to choose the worker with matching qualifications. The quality of a crowdsourced task's outcome is affected by workers' quality: worker's general reputation and experience [34].

The reputation is a public metric and community based, while experience is task dependent. The reputation can be measured by rating the quality or timeliness of worker submitted work or as a feedback from the community. AMT only allows workers who match the required reputation levels; e.g., only workers with a specified percentage of accepted works or living in certain areas can participate [33]. Workers' reputation is used in [42] to assess credibility of the contributed data.

On the other hand, experience is the knowledge and skills a worker gained while working in the system or through support and training. To judge the level of expertise of a worker, credentials are needed; e.g.: academic certificates or degrees, spoken languages, or geographical regions that a worker is familiar with can be credentials. In Wikipedia, only workers with required credentials can participate [33].

- *Evaluation based on ground truth* In [44], the problem considered is counting the nouns in a word list. The workers are selected randomly from a set of workers who have 100 or more approved AMT HITs with one 95% HIT approval rate. They earn bonus, varying from low, medium to high, if their answers differ by a maximum of 3 words from their peer (previous worker) answer. The label accuracy is decided using Eq. 1.

$$\text{Label Accuracy} = 1 - \frac{|A_s - A_w|}{\text{Max}(\text{HV} - A_s, A_w)}, \tag{1}$$

  where A_s is the standard answer coming from the gold standard or the peer answer, A_w is the worker answer, HV is the highest possible value for the answer. The accuracy is 1 for the best answer matching the standard and 0 for the worst answer. *The worker's quality is calculated by averaging the accuracy of five labels generated by him.*
- *Evaluation without ground truth* In [40], the authors presented an expectation maximization algorithm (EM), using maximum likelihood, for inferring the error rates of annotators that

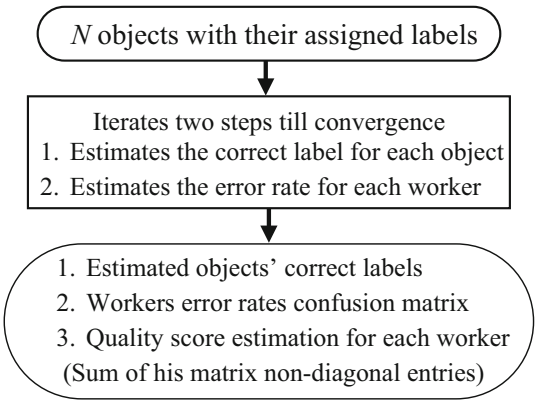

Fig. 2 EM algorithm flowchart

Initialize the error rates π_{ij}^{k} per worker k as perfect worker
Initialize the correct label per object $T(o_n)$; use majority vote
While not converged do
- Estimate the correct label $T(o_n)$ for each object, using the labels assigned to object o_n by workers; weighting votes by each worker error rates π_{ij}^{k}
- Update the error rates π_{ij}^{k} , for each worker k, using the estimated correct labels $T(o_n)$ and workers assigned ones
- Estimate the class priors $Pr\{C\}$ for each class C

end
Return the estimated error rates π_{ij}^{k} , the estimated correct labels $T(o_n)$, the estimated class priors $Pr\{C\}$

Fig. 3 EM algorithm as in [35]

assign class labels to objects, when the ground truth is unknown. They calculate the error rate for each annotator based on his performance history.

This rate is used as a weight for calculating a weighted consensus among workers labels to decide the correct label instead of using a simple majority opinion. Each object submitted to the EM algorithm is annotated by one or more of a K varying quality workers. To measure the worker quality, the algorithm assigns each worker k a confusion matrix which gives the probability that worker k will mistakenly label a class i object as a class j object; further details on EM are in Sect. 2.4.2.

The EM algorithm is used to quantify worker quality in [35]. It takes the set of labels assigned by each worker to N objects and outputs a confusion matrix with each matrix element representing the quality of each worker associated with each label class. The algorithm infers the correct label for each object, see Figs. 2 and 3.

It is noticed that to find the true quality of the worker with high degree of accuracy, approximately 20–30 labels from each worker are needed for objects with 5 labels each. It is also found that the number of labels/worker affects estimation quality much more than the number of labels/object [35].

The disadvantage of EM is that it does not consider systematic bias. Therefore, a perfect worker who by mistake flips all answers will have 100% error rate, while a spammer who always gives the same answer ends up with 50% error rate. The errors in the first case are reversible, while they are not in the second.

The algorithm presented in [35] separates the unrecoverable errors from those caused by bias; errors caused by biased workers are reversed to higher-quality label assignments. This correction happens by transforming the label assigned to each object by each worker using the error rate of this worker into a new label, which is the best possible estimate available for the true label assignment. So, if there are L possible classes and worker k assigns an object class j label, this label is transformed to a new label by multiplying the worker error rate by class priors, see Eq. 2, then normalize using the summation of all class priors as in Eq. 3:

$$< \pi^k_{1j} \cdot \Pr\{C = 1\}, \ldots, \pi^k_{Lj} \cdot \Pr\{C = L\} > \tag{2}$$

$$\text{Prior}\{\text{AC} = j\} = \sum_{i}^{L} \pi^k_{ij} \cdot \Pr\{C = i\}. \tag{3}$$

The transformation cost is used to differentiate the perfect worker from the spammer. To estimate this cost, the misclassification costs C_{ij}, incurred when an object of class i is misclassified as j, are used; C_{ij} is 1 for misclassified objects and 0 otherwise. Given a set of classification costs C_{ij}, and a new label $p = \langle p_1, p_2 \ldots, p_L \rangle$, the new label expected cost is:

$$\text{Cost} = \sum_{i=1}^{L} \sum_{j=1}^{L} p_i \cdot p_j \cdot C_{ij}. \tag{4}$$

p_i is the probability of the object to be in class i. Cost is 0 for perfect workers and high cost is assigned to spammers; low costs are assigned to reversible errors not only for correct answers. This is the way the system separates low-quality workers from high-quality or biased workers. One way to decide the threshold cost that differentiates the spammers from non-spammers is to find the expected cost of a worker who always assigns the same label and the assignment resulting in the minimum expected cost should be used as threshold [35].

- *Context-based evaluation* In [5], the authors based their worker trust evaluation on contextual information relevant to crowdsourcing since available context-aware trust evaluation models are not readily applicable to crowdsourcing. Earlier work in trust management relied on manual monitoring and blocking workers to prevent untrustworthy behaviors without clear criteria for identifying untrustworthiness. Although later work added verification questions to improve the quality of answers, again no criteria is found for determining the difficulty or frequency of these questions. Even existing trust management methods adjusted to fit the crowdsourcing environment still assume that the worker has equal ability to solve all types of tasks which is not true; trust should vary with the task and context. In addition, cheating detection usually happens after answers submission. Selecting trustworthy workers and avoiding malicious ones before answer submission improve answers quality and prevents wasting resources.

Due to all mentioned limitations of workers' trustworthiness evaluation methods, a two-dimensional context-aware trust evaluation model for finding qualified workers for the considered task is proposed in [5]. The authors considered two factors that affect worker performance: *task type*, performance can be satisfactory only in worker's familiar tasks, and *the reward amounts*, high performance in certain reward range indicates trustworthiness in tasks within similar reward ranges. Both factors are used to reach two types of context-aware trust: *task* and *reward* based. The tasks are classified according to their *input* (figural, symbolic, semantic, audio, video), *processing* (cognition, memory, divergent production, convergent production and evaluation) and *output* (units, classes, relations, system, transformations and implications) which are represented in three-dimensional intelligence spaces.

In this space, the worker historical records are represented with a Trust Cube (TC) with *ha* representing the number of approved tasks, *hs* is the number of submitted ones, and *hr* or the approval rate which is equal to (*ha*/*hs*). When a worker applies for a task, the value of the worker's task-based trust is differently influenced by his TC records. The values of the influence factor range from 0 (worker's approval rate in TC_j is irrelevant to trustworthiness) to 1 (the influence is highest). The *task* trust value relies on *ha* and *hs* values and the influence between tasks; the more the dimensions shared between TCs, the higher the influence since this reflects higher similarity between the two tasks.

For the *reward*-based trust, a worker high performance in a reward range indicates that the worker is trustworthy to handle tasks in the same reward range. Changing the reward value changes the performance. Usually, the final worker trustworthiness value is calculated using a normalized weighted average of considered factors, in this case the task- and reward-based trusts, but to avoid the weights subjective bias they handled the trustworthiness selection problem as a multi-objective combinatorial optimization problem without subjective weights; best worker combination of average task- and reward-based trusts is achieved when none of these two can be improved without degrading the other.

In [19], another application for context-based participants selection is studied. The application considered is mobile sensing which uses crowdsourcing to obtain information from mobile users. It is found that the quality of users' reports is affected by their attributes; therefore, in their SACRM system, they considered three factors for participants' selection:

1. *Social attributes* (interests, friend circle, living area)
2. *Expected delay* (time between task assignment and response) which is crucial for delay-sensitive tasks and relies on user location and time availability.
3. *Reputation* (quality and reliability of users reported data) a reputation database is created to record users' reputation and initialize it for newly registered users

SACRM main components are:

1. Participants selection
2. Assessment and rewarding of sensing reports; report quality and delay are the report assessment metrics
3. Reputation Management

The data requester asks for participants to contribute to task *t* with requirements including the needed social attributes set, the maximum delay and the task budget. To choose the best participants for this task, a utility function for each attribute is used such that:

1. The number of common social attributes between the participant and task *t* requirements list divided by the number of task required attributes should be >0
2. The user delay $\leq$ max delay required by the task
3. The user reputation should be within the reputation range preset in the task requirements

After defining utility functions to quantify the effect of each of the three considered factors (social attributes, delay and reputation) on crowdsourcing, they formulate the participant selection problem as a combination optimization problem, so they only choose a subset of participants who can maximize total utility *E for task t*.

Maximize:

$$E_t = \sum_{j=1}^{n} e_t^j x_j. \tag{5}$$

Subject to:

$$E_t = \sum_{j=1}^{n} b_t^j x_j \leq \text{bid price}, \tag{6}$$

where e_t^j is the utility *and* b_t^j is the bid price of candidate j for task t, $x_j = 0$ or 1 shows whether the candidate j should be chosen to participate ($x_j = 1$) or not ($x_j = 0$). The 0 is used to exclude the ones with the expected delay larger than the max or the bid price larger than task bid. The focus is not only on trustworthiness but also on participants bid price. This price is considered in their reputation to evaluate their cost performance ratio; participants finishing sensing tasks with same quality but lower bid prices should have higher reputation. *Quality assessment and rewarding scheme rely mainly on level of accuracy and delay of sensing reports*. *Accuracy* is reflected in the level of supports and conflicts the submitted reports obtain from other sensing reports. Having similarity score of -1 means completely conflicting, and 1 means exactly consistent with the other reports. The higher the similarity scores, the higher the quality score of the submitted reports; the effect is decreased with fewer reports. For *delay* deviation assessment, they have a utility function based on how far the actual delay is from the expected one; the higher the deviation the lower the score. The reward is assigned based on assessment; if the assessment score is below the threshold decided by the task requirements, the sensing report is identified as poor otherwise its user is rewarded by the user bid price. The final utility is the sum of assessed quality scores; see Eqs. 5 and 6.

2.2.2 Task QC

Quality control is also crucial at task design time and runtime since mistakes can happen even with high-quality workers. Hence, there is a need to define, measure, and manage task quality in crowdsourcing. Currently, requesters rely on ready techniques embedded in host systems that are not customizable. Using tools like TurKit for QC processes requires programming skills. Hence, customizable robust crowdsourcing QC methods are needed. In [33], the following techniques are listed.

1. *Quality of the task design* effective preparation of tasks with defensive design and unambiguous description explaining evaluation and compensation are needed to avoid cheating and enhance quality.
2. *User interface* friendly interface encourages more participants and helps lift quality; should be complex enough to filter out spammers by verification questions and simple enough to avoid delay.
3. *Granularity* simple tasks such as tagging are usually short and need little expertise. Conversely, complex tasks such as writing an article need to be broken into subtasks since solving it by one person is time- and resource-consuming, it needs expertise that might be difficult to find. Therefore, workers solve the subtasks, and their contributions are consolidated to build the final answer. The quality is affected by how well the contributions are integrated.
4. *Compensation policy* disclosure of incentives to the workers may result in getting higher quality.

Good task design and only allowing trustworthy people to participate may increase the outcome quality, but cannot cancel the importance of applying QC at task runtime, while collecting data and when aggregating contributions. During runtime, workers should be provided with real-time support to lift their contribution quality. For complex tasks, the

workflow management should be applied to have quick QC approaches. In the following subsections, QC approaches applicable during data collection and contributions aggregation are discussed.

Crowdturfing[1] tasks are defined as a combination of crowdsourcing and astroturfing.[2] After analyzing abusive tasks by [31], it is found that 90% of the crowdsourcing tasks on sites like Freelancer and other Chinese sites were crowdturfing tasks [11, 31]. After analyzing western crowdsourcing sites and Twitter by [11, 22], it is found that malicious tasks in crowdsourcing systems target either online social networks or search engines. It is found in [11] that Fiverr micro-task marketplace has two types of sellers: legitimate sellers and unethical (malicious) sellers who post crowdturfing gigs; this is more than 50% of Fiverr gigs, which manipulate targeted sites such as online social networks and search engines. These gigs are clearly used to provide an unfair advantage for their buyers; e.g., "I will provide 2000? Perfect looking twitter followers."

Crowdturfing tasks have become the most popular as a result of sellers and buyers abuse of the micro-task marketplace. This degrades information trustworthiness on the entire Web. In [23], ways to analyze, detect and remove these crowdturfing tasks and prohibiting sellers from posting those gigs are discussed. The most interesting features are category features: Gigs are categorized into top and 2nd level, a world domination rate (number of countries where buyers of the gig were from, divided by the total number of countries), and bag-of-words features such as "link," "backlink," "follow," "twitter," "rank," "traffic" and "bookmark."

Task design and how they affect the accuracy of the resulting relevance of labels are discussed in [23], the effect on workers and on the quality of the collected data.

For example, questions needing investigation include whether the quality of output increases with pay and whether the workers who are motivated by fun would be put off by higher pay. In [1], the authors investigated how varying the HIT design, e.g., title, terminology, pay, affected annotation accuracy. In [6], the authors studied the effects of different social and financial incentive schemes, but found that results were mainly dependent on the task difficulty.

In [9], the authors investigate both the effects of task design parameters, including the offered pay, effort, and the human factors, such motivation for accepting an HIT or their satisfaction with their pay. They considered three key attributes of a crowdsourcing task: pay, effort and qualifying criteria. For each of the three attributes, they investigate two settings corresponding to 8 distinct task conditions:

- Pay—they experiment with 2 levels \$0.10 or \$0.25/HIT.
- Effort—they varied the effort required to complete the task through the number of pages included in a HIT: 5 or 10 pages per HIT.
- Qualifying criteria—they leverage AMT's worker pre-filtering feature that incorporates worker reputation measures and use two settings: open call where no qualifying criteria is required from workers to gain access to the HITs and restricting access to workers with > 95% HIT approval rate and > 100 approved HITs.

They found more accurate labels with the higher pay, where more accurate workers perform more HITs. It is also found that the higher the familiarity the less the accuracy. No

[1] Crowdturfing uses crowdsourcing platforms to spread biased opinions and framed information through malicious URLs in social media, forming astroturfing[2] campaigns and manipulating search engines, which degrades the quality and usefulness of online information[25].

[2] Astroturfing is the campaign that masks its supporters and sponsors to make it appear to be launched by grassroots participants [25].

relation is found between the reported levels of task difficulty. The relation between workers' satisfaction with pay and HIT accuracy is significant over all HITs.

2.2.3 Data QC

Since even with high-quality tasks low-quality contributions can still exist due to mistakes or misunderstanding, QC approaches should still be used during the data collection stage to have reliable answers [33]. Crowd data can be discarded by scientists due to lack of quality information; therefore, the ability to verify and validate data collected from participants is crucial [42]. In [42], the authors used QC pillars for concerns occurring while acquiring crowd data from mobile handheld devices. They came up with a generic quality assurance framework and data quality standards including internal quality (completeness, consistency, attribute and positional accuracy) and external quality (fitness for use).

The quality elements used to select high-quality data in [30, 36, 42] are:

Vagueness: lack of classifying capability which is related to the confusion levels mentioned in [36].
Ambiguity: lack of understanding or clarity like sentiment ambiguity mentioned in [36].
Judgment: accuracy of choice in relevance to facts which is also mentioned in [30].
Reliability: consistency in choices.
Validity: coherence against other knowledge relevant to the noise in [36] and relevant to validity of objects, data types, or links in [30].
Trust: confidence based on contributions or reputation

In [36], the authors also added:

Noise level: deviation from the gold standard

Confusion: lexical uncertainty confusion

In [30], the authors listed data quality elements applicable to linked data quality assessment for the DBpedia application such as interlinking and representational consistency relevancy.

2.3 Verification methods

- *Rewards and penalties* In [14], the participants can recruit others as answerers and they become responsible for verifying the answers returned by those recruited answerers. They then have to pay a *penalty* in case false answers are propagated back to the one who originated the question. In [41], three ways are discussed to enhance quality against worker errors. They recommended the use of more workers, giving monetary *bonuses* to quality annotators beside penalizing unreliable ones, and modeling the reliability and biases of individual workers to correct them.
- *Redundancy* Getting multiple annotations per object reduces the influence of occasional errors and catches malicious users. The more the annotations, the more expensive the process [3]. Using majority consensus, control group as in [42] or answer agreement among independent workers for the same input as in ESP game proves correctness [33]. Furthermore, input description agreement among workers is accepted as a quality answer (e.g., Tag-A-Tune) [33]. Also, majority decision among reviewers judgment on the contribution's quality is taken as its actual quality (e.g., AMT). It is stated in [18] that in the crowd labeling settings, one common QC approach used by employers is to introduce redundancy which is very similar to majority voting (MV). The worker accuracy rate is measured by

the proportion of labels submitted by the worker in agreement with the majority label. The drawback is that it ignores any heterogeneity in workers quality and discards trustworthy ones whose answers are wrong but correctable [18].

- *Manual verification* This can be done if the question has countable results like: the DARPA application in [28], finding individuals like natural disaster survivors, crime suspects in certain areas [14] or specialists to assist in the area where the disaster took place [13]. In addition, explicit quality assessments can be done by a separate low-cost easy grading task where a worker looks at several annotated objects and gives a score per annotation to ensure the quality of collected annotations [3]. In [16], the authors use the AMT service for labeling vision data and to ensure highly accurate dataset; after they collect the labels for the set of images through crowdsourcing, they verify the collected labels by humans to make sure they truly belong to the synonym set chosen by the annotators.
- *Automatic validation* using computational techniques to do credibility check to the data collected (e.g., automatic correction of spelling) [42].
- *Comparison against authoritative data* To ensure best worker performance and to check quality, a cheap strategy is followed where a collection of objects, e.g., images, is prepared with only a fraction coming from a set with trusted annotations (the gold standard set) to keep the cost low. The worker does not know if an object comes from the new data or from the gold standard set while providing the annotations. If his annotations or answers significantly deviate from the gold standard (known answers or common sense facts), the worker is suspected of not following the instructions. Therefore, the gold standard annotation is revealed to the worker after his submission. This immediate feedback reminds and encourages workers to follow the system protocol [3]. In [42], the authors also assess validity against authoritative data by comparing collected data with authoritative data to improve the confidence and validity of collected data [42]. This is also used in AMT [33].

The drawback of this approach as mentioned in [18] is that additional cost is incurred in labeling standard data. In addition, this method discards low-quality work which can be combined to provide high-quality outcomes [33].

- *Model-based validation* focuses on comparison of the crowd data with data from models or previously validated crowdsourced data. For example, using environmental models, it assesses the discrepancy between crowd inputs and model predictions [42].
- *Pairwise comparison* In [48], it is mentioned that the pairwise comparison is simpler for the crowd and has higher accuracy than rating methods. The pairwise comparison is utilized in [48] to decide the top K elements using rounds of pairwise comparisons. For efficiency, not all the possible pairs are presented to the crowd, an iterative method is used instead where a subset of pairs are crowdsourced in each round, and based on the round results the next round pairs are decided. For result inference, they used heuristic-based algorithms and machine learning methods. In [3], the peer consistency is the metric used to ensure selecting the best annotation.

It is recommended to calculate the consistency scores for a pair of annotations ($a1$ and $a2$) of the same type by calculating XOR($a1, a2$)/OR($a1, a2$), where the XOR measures how far the two annotations are and the OR measures their closeness, and then selecting the pair with the lowest score representing the most consistent pair as the best annotation for that object or image [3]. The peer consistency evaluation is also used in [44] where the authors found that using this type of consistency checking can lead to better results than evaluating workers contribution against the gold standards.

In [20], the main task is to get the best n translations of a sentence from a set of crowd answers. The authors use tournament selection [49] and associate probabilities reflecting each candidate quality.

The tournament selection in [49] takes place after collecting n results of HCTs computed by n independent candidates and discarding duplicate results such that only k unique answers are left ($k < n$). In the tournament selection step described in [49], two results are picked randomly from the k answers and then a person gives an up or down vote on the comparison; the result that is up-voted goes into the pool of "next generation"; this step is repeated n times to generate a pool of size n answers.

These steps are applied to the translation problem in [20]. Pairs of candidate translations are presented to crowd workers to decide which one is a better translation of the source sentence; *n* workers produce *n* translations that are the best in the bigger set of translations. This can be repeated till the stopping condition is met which can be reaching a size that is 20 or 30% of the original set which needs nearly 5 rounds. The majority answer is then picked from the winning set.

- *Employ experts as validators* Domain experts can be employed to review, check and evaluate quality of collected data as in Wikipedia [33]. Domain experts can be consulted to validate an observation if required [42].
- *LBS (Location-Based Service) positioning* where location data are used to control software features. It ensures that data are collected from the right positions and avoid data coming from inaccurate ones. It also customizes options presented to users based on their location to simplify their tasks and reduce errors [42].
- *Cleaning data* happens by rejecting malicious contributions or refining them for subsequent processing, e.g., removing data captured from geographic positions outside the study area and stop word removal [42].
- *Linked data analysis* Publicly available feeds such as Twitter are employed as a reference to newly captured information. A conjunction of assessments from authoritative comparison and linked data analysis can enhance the quality of captured data. Some data can be deduced to have low quality when assessed by authoritative data, but being tweeted a lot in the same time frame and location can strengthen its validity [42].

2.4 Information aggregation

The collected information from users should be aggregated to find a meaningful output. For example, in a crowdsourcing task of comparing instances to choose the best, tasks are presented to people iteratively to reduce the number of participants. For 1000 objects, 100 participants can be recruited with 10 instances each and then the resulting 100 instances can then be presented to 10 more participants. To increase accuracy, more participants will be needed per instance and their contributions should be aggregated to reach one answer per instance. This leads to facing quality–cost–latency trade-offs; the more participants per instance, the higher the cost and latency [39].

Some problems need very simple aggregation like the DARPA Balloon finding problem mentioned in [28] where participants are asked to retrieve the coordinates of ten balloons placed in various locations in the USA. The output of such problem is the coordinates of ten locations, and no aggregation is required in this case. In the case of paired games utilized for obtaining annotations that can be used for labeling images or for detecting objects in security systems [26], there are many algorithms for annotation collaboration. To reach the most precise set of labels for the annotated dataset, the aggregation happens by using two

or more players who should agree on the same label per object to be able to win the game; only agreed upon labels are considered as valid ones. Other problems need higher level of collaboration and more complicated methods. For example, in instance labeling applications as mentioned in [37] not all annotators have the same level of experience and some completely misunderstand the task at hand. Other annotators provide any labels randomly in hope that this will not be noticed and they get paid. Therefore, each instance is annotated by more than one person and then a way to collaborate the annotations should be found.

The main aggregation approaches are summarized in this section. According to [34], these techniques are broadly classified into two categories according to their computing model: non-iterative, uses heuristics to compute a single aggregated value of each question separately, and iterative, which contains rounds of probability estimation for possible labels of each object.

2.4.1 Non-iterative methods

In [34], three non-iterative algorithms are discussed: The popular and simple majority decision (MD) algorithm does not require preprocessing; the answer with highest votes is selected as the final aggregated value, very similar to the popularity rule (PR) in [39]. Then, the honeypot (HP) approach filters spammers' answers in advance and the Expert Label Injected Crowd Estimation (ELICE) considers worker expertise and question difficulty [34].

- *Majority Decision (MD)*, sometimes called majority voting (MV) [34, 38] or majority consensus [37], is a straightforward method that aggregates each object independently. In [33], given an object o_i, among k received answers, the number of answers for each possible label lz is counted. The probability the true label is lz is the percentage of its count over k. The MD drawback is that it does not consider the workers varying levels of expertise which is a problem if most of them are spammers [34].

In [38], more than one annotation is required for each instance and the *majority voting* is used to overcome lack of reliability of some annotators. In [36], snippets are taken from election blogs with text localized around presidential candidates. The elections blogs snippets are annotated by non-experts with the objective of using these annotated instances in building a classifier to detect positive and negative mentions of presidential candidates. Since annotators are not experts, multiple participants are required to label the same snippet to have reliable annotation. The winning label becomes the gold standard label of a snippet which is the label that receives the majority of votes. Snippets whose annotations fail to reach majority votes by annotators are excluded.

In [37], the authors also obtain ground truth annotations from crowdsourced labels by applying a majority consensus heuristic. In [41], multiple-choice responses or numeric input within a fixed range is considered which simplifies the aggregation. Ten independent annotations per item are collected, and simple aggregation methods like averaging are applied to improve the quality and reliability of the annotations. In [3], the aggregation is done by selecting the most consistent pair of annotations as the best annotation for the object.

- *Plurality Rule (RP)*, very similar to MD, explains the case where more than one label has equal votes. The plurality rule states that if a set of z labels each got n responses and no label had more responses than n, then these labels are called the winners and any of them can be the maximum. One of them can be picked at random as a valid representative of the object [39].
- *Honeypot (HP)* operates as MD except for filtering out untrustworthy workers in a preprocessing step. HP merges a set of trapping questions (whose true answer is already known)

into the set of questions randomly. If workers fail to answer a specified number of these trapping questions, they are classified as spammers and blocked. Then, the probability that a possible label be assigned to an object is computed using MD among the rest of workers. The drawback of HP is that the trapping questions might be subjectively constructed and not always available; too difficult ones can lead to blocking truthful workers as spammers [34].

- *Expert Label Injected Crowd Estimation (ELICE)* extends HP by considering the trapping question difficulty as well as using it to estimate workers expertise; nevertheless, ELICE also has the same drawback with regard to the availability of the trapping set [34]. The question difficulty is estimated by the expected number of workers who correctly answer a specified number of trapping questions. In addition, workers level of expertise is represented by the ratio of their truly answered trapping questions to the total number of question. Finally, using logistic regression, the probability that the object aggregated label is equal to one of the possible labels is calculated given all workers contributions; each answer is weighted by worker expertise and question difficulty.

2.4.2 Iterative methods

On the other hand, there are iterative aggregation algorithms listed in [34]: the message passing (MP) in [18] and the expectation maximization (EM), supervised learning from multiple experts (SLME), generative model of labels, abilities, and difficulties (GLAD), and iterative learning (ITER) in [34], each consisting of two updating steps repeated till convergence:

1. Update the aggregated value of each question based on the expertise of the answerer.
2. Adjust the answerer expertise based on his answer compared to the aggregated one found in step 1.

Message passing (MP) infers the true classes of objects and the reliability of workers. Higher weights are given to labels coming from trustworthy workers. The initial worker reliability is assumed from a Gaussian distribution N (1, 1), and then the estimated correct label is found by multiplying each worker answer by this initial reliability. The initial reliability is then modified at each object update: The worker label is compared to the estimated correct label, and the reliability of the worker is then updated depending on how close the worker label is to the estimated one; the closer it is, the higher the confidence should be in that worker. Cycles of updates happen till convergence, and the output is the estimated set of labels and the workers adjusted reliability [17].
Expectation Maximization (EM) is an iterative algorithm that aggregates many objects at the same time. The running time for the algorithm is high since it takes a lot of steps to converge. To estimate the class probability of each object, the algorithm iterates two main steps, expectation (E) and maximization (M), until convergence when the probabilities stop changing. As discussed in Sect. 2.2.1, the performance of the workers and their accuracy estimates are represented in a confusion matrix which lists the probabilities of different classification errors made by each worker. In [17], each element in the matrix represents the probability that worker k does the error of misclassifying an object of class i into class j.

The algorithm iterates two steps until convergence:

(1) The (E) step estimates the true class for each object using the labels provided by workers; it accounts for workers error rates by weighting their answers with their current estimates of expertise or error rates [34].

(2) The (M) step re-estimates the error rates [18] or expertise of each worker [34] by comparing the worker submitted labels with the estimated true class for each object (the output of the first or E step).

As mentioned in [34], the EM algorithm takes as input the set of labels L and class priors and outputs the following:

1. The class probability estimates for each object: The probability the object belongs to class C_i is calculated by dividing the number of workers who labeled the object to belong to class C_i over the total number of workers who labeled the object
2. Confusion matrix for each worker (k) where each element e^k_{ij} is equal to class i probability * number of times worker k mistakenly chose j instead of i
3. Class prior estimates (number of objects classified as class i divided by the total number of objects).

Section 2.4.2 has further details of the EM algorithm.

- *Supervised Learning from Multiple Experts (SLME)* operates as EM without confusion matrix. It characterizes the worker expertise using two statistical measures, sensitivity and specificity. Sensitivity is the ratio of positive answers which are correctly assigned, while specificity is the ratio of negative answers which are correctly assigned. SLME can only be used with binary labels, and it is not suitable for multiple labels as MD, HP and EM [34].
- *Generative Model of Labels, Abilities, and Difficulties (GLAD)* extends the EM algorithm [34]. In [34], the authors noticed that for the same question the expert answer is better than that of the non-expert, while the expert performance might degrade as the difficulty level of the questions increases. Since previous iterative algorithms concentrate on the effect of worker quality on their contributions quality, the GLAD algorithm came to draw the attention to the fact that the varying difficulty of the questions can also affect answers quality [34, 42]. The EM approach is used to obtain the estimates of worker quality, the difficulty of each question and the true class of each object.

1. The *easiness* of all objects is initialized to 1,
2. The *class probability estimates for each object,* or the object's probability to be in class i, is initialized by the number of workers classifying the object in class i over the number of those labeling the object.
3. The *class priors* are estimated by adding the probability of each object belonging to the considered class (the class probability of each object) and then dividing this by the total number of objects.

- *Iterative Learning (ITER)* is similar to GLAD algorithm in being iterative and considering the worker expertise and question difficulty. The basic difference though is that the worker contributions are not handled as one value represented in worker error rate or expertise. The worker reliability is handled in relation to each question and the question difficulty is also decided with respect to the answerer [34].

The final value for worker quality is the sum of the worker reliability per answer weighted by the question difficulty level associated with that worker which makes ITER insensitive to the initializations of worker reliability or question difficulty [34].

2.5 Topical experts discovery

One way to ensure validity of gathered information is to rely on reliable sources such as experts in the topic considered and only consider answers coming from them. According to [12], accomplishing a task may require the expertise of multiple actors, and harnessing that expertise requires identifying who the experts are. In [37], it is emphasized that having unreliable annotators causes the noisy labels problem which results from the need to label each instance by many different annotators hoping to have good labels among them or reach a consensus from large number of labels. Although the cost per label is low, this cost accumulates rapidly when searching for a given label quality and increasing the number of annotators hoping to reach it. However, if an expert annotator provides a label, one can probably rely on it being of high quality and may not need more labels for that particular task. Therefore, utilizing experts reduces the number of labels required which lowers the total labeling cost as well as the error rates.

In [38], the authors came up with a model that provides more reliable labels with the minimum number of annotators. They model annotators and image difficulty as multidimensional quantities where each annotator is described by three attributes, competence, expertise and their bias that reflects how users weigh errors differently. Their annotation noise is measured by the annotator deviation from the true label; the more noise the annotator has, the more inconsistent their labels will be. The image ambiguity is also considered since it can lead to inconsistent labels regardless of the annotator skills. The deviation of the annotator from the ground truth annotation is considered noise and noisy annotation results in inconsistent labels.

It is found that the more competent and experienced the annotators are, the lower the variance of the noise. In [36], the noise level represents the deviation from the gold standard labels. The noise level of a particular annotator is then calculated by adding the deviation of his annotations from the gold standard.

The noisier the annotators are, the lower the mean agreement with gold standard. Therefore, it can be concluded that the more experienced the annotators are, the higher the resulting accuracy.

Many ways exist for discovering topical experts. They can be identified using heuristics, referral, through analyzing user social media records and contributions as follows:

- *Heuristics* In [41], it is elaborated that AMT allows a requester to restrict which workers are allowed to annotate a task by requiring that all workers have a particular set of qualifications, such as sufficient accuracy on a small test set or a minimum percentage of previously accepted submissions. In [37], the ImageNet hierarchical image labeling model uses an algorithm that finds and prioritizes experts when requesting labels and actively excludes unreliable annotators. It categorizes images in a semantic hierarchy of labeled images consisting of 12 subtrees representing the main categories under which images are classified hierarchically. Each subtree contains the synonym sets that group all images sharing the parent common concepts. They developed online crowdsourcing labeling algorithm that relies on rating annotators to obtain cost-effective labels. Based on labels already obtained, expertise and reliability are assessed at runtime by this online algorithm. The algorithm iterates two steps until all the images are labeled: (1) labels collection step, (2) annotator evaluation step. It dynamically decides the number of labels and labelers needed to achieve a desired level of confidence and reach consensus on labels. The drawback is that the discovery cannot be done in advance, and it is done during runtime. Therefore, other

forms of expert discovery are needed to allow finding experts prior to dispatching the task so it only reaches trusted answerers.

- *Referral* In [47], a type of crowdsourcing called referral-based crowdsourcing is used where information requests are propagated recursively through invitations among members of a social network. Each participant is encouraged to recruit more experienced candidates that he knows to be more capable of solving the problem than him. They utilize the social network for wide dissemination of the task, and then incentives are given to people for either their own participation or for their recruitment of more experienced answerers.

A recursive incentive mechanism is used to reward the answerer as well as all those along the referral path that led to the answerer so that if the person who finds the right answer gets C compensation, the immediate recruiter gets $C/2$, etc. This is called the split contracts which improve the older query incentive networks method used for favor exchange over social networks; recruiters offer incentive B to convince others to join in. The B value keeps the recruitment chain growing until finding an answer. At least two friends should be convinced which is impractical; therefore, the split contracts with its below 2 branching solve this problem [29]. Since the recruitment is based on the recruiter belief in the recruited person ability to answer, it should ensure that only people with high probability of being experts are recruited. Unfortunately, malicious candidates exist in social networks. To minimize incidents of malicious participants, the recruiter should verify the answers coming from the recruited person and be penalized for wrong answers propagated to the source. The solution still need to be tested in practice; the game-theoretic analysis shows that penalizing intruders does not necessarily stop them, on the contrary, they might become more aggressive [29]. More sophisticated methods of expert discovery are needed to truly differentiate between true experts and intruders.

- *Analyzing User Records on Social Media* The users' accounts on social media can be analyzed to extract information that can help in identifying expertise in different topics. In ImageNet and similar applications that do not have enough information about the user, experts cannot be discovered beforehand. They are discovered dynamically as more people join. In contrast, in the case of the social media and Twitter information about users is readily available in their accounts and contributions. Therefore, the expert discovery can be done beforehand to reduce runtime processing, the number of labels per instance and the overall cost. Users' profiles, bio, followers' information as well as their contributions and message content can be mined to extract valuable information that helps identify topical experts.

The work in [43] utilizes Twitter lists as a main source for finding experts, while the work in [2] considers fifteen features extracted from Twitter social graph and tweets content to identify experts which might be biased by the content of the tweets at the time of evaluation. In addition, the Twitter official who to follow (WTF) service [46] uses profile information such as user name and bio, social links, and local engagements to decide on user expertise in a certain topic.

Since the two methods in [2, 46] rely on features where the users can freely express themselves, they can be biased by the content of those features, such as the bio, in case some users include mocks in the content [43]. A Twitter user can organize other twitterers who are knowledgeable in certain topics in lists and then view aggregated tweets posted by these users in the List timeline. Researchers in [43] had the potential of using these lists as well as their metadata such as list name and description to find experts. They managed to extract information from the crowdsourced lists to build search engine for finding topical experts on Twitter.

The method used can be summarized as follows:

1. Gather crowd-created lists for all Twitter users
2. Mine list metadata to infer the users' topical expertise
    a. Each list has a name and description (metadata) and a list of members. Extract the frequent words from lists metadata and deduce them as topics for listed users
        i. Separate CamelCase words into individual words.
        ii. Do case-folding, stemming, and filter out stop words including domain-specific ones (Twitter, List, formulist).
        iii. Identify nouns and adjectives using a part-of-speech tagger.
        iv. Group similar words across languages close to each other based on edit-distance (politics/politica)
        v. Topics with 1 or 2 words maximum are used for each user; the result is a topic vector per user $\{(topic_1, fr_1), (topic_2, fr_2), \ldots, (topic_n, fr_3)\}$ where fr is the frequency in list metadata.

2.6 Ranking experts

After finding the experts, it is better to rank them with the level of expertise to decide their priority and the weight for their answers validity. The users' expertise should be ranked according to its relevance to the query topic. Ranking in Twitter relies on measures extracted from the social graph and tweet content [46], while ranking in [2] is based on studying fifteen Twitter features. Ranking in [43] is based on the Twitter list. It computes topic similarity score between user topic and query vectors using the cover density ranking algorithm. This is best for queries with 1 to 3 terms [43]. Cosine similarity will not be effective because of short queries, while the cover density measure ranking is based on term proximity and co-occurrence [7]. Finally, the topical similarity score for a user is multiplied by the log of Lists number containing the user; a user included in more Lists by others is likely to be popular in Twitter.

The Cognos system in [43] is evaluated by selecting well-known expertise for the set of test topics, and for each topic a benchmark top expert lists are obtained from Twitter WTF service [46] and the state-of-the-art research systems for identifying topical authorities [2]. It is discovered that List metadata is often sufficient to infer high-quality expertise for a variety of users. Cognos proved to yield better search results, especially in the cases where the bio or tweets posted by a user do not correspond to or contain information about the user's topic of expertise. Furthermore, Cognos is more efficient since it performs as good as or better than the official Twitter WTF service for more than 52% of the test queries, even though it is based on a single and simple feature (Lists) [43]. The mean average precision of the top 10 results returned by Cognos is above 0.9.

To reduce problem size, they used a maximum limit of 2000 lists per user because of the rate limitations in accessing the Twitter API. The evaluating users are asked to give binary feedback on the relevance of the top 10 results returned for each of the chosen 55 test queries. Their accurately predicted relevance was 75% and above [43]. For updating the system with new users and to limit the size of crawled user accounts, they decided to only crawl hubs of users that follow large number of experts and include them in their lists. Top hubs can be discovered using HITS algorithm. These hubs discover experts who join Twitter and add them to their lists. The proposed Cognos system then crawls their lists to get information

about the experts. Crawling the top million hubs takes 3 weeks; therefore, monthly updates are decided.

In [17], the authors decided to identify Twitter experts in certain topics and rank them based on their topic-specific TwitterRank. This rank is proposed to measure the topic-sensitive influence of the twitterers. Twitterers's general influence can be measured as an aggregation of the topic-specific TwitterRank in different topics. To make sure that the effect of the reciprocal followers is minimized while measuring the influence of the followed person, the transition probability matrix should consider the topical similarity between the connected twitterers. Topics of interest to twitterers are extracted from analyzing their tweets content. Topic-specific relationship networks among twitterers are constructed. The used TwitterRank algorithm for topic-sensitive user influence ranking considers the twitterers topical similarity as well as the link structure when calculating the influence. Each twitterer is represented by a document that includes all his published tweets to avoid single tweet bias. The latent Dirichlet allocation (LDA) model is used to represent each of these documents as a probability distribution over some topics. This distribution is represented as a $D \times T$ matrix, where the D is the number of twitterers and T is the number of topics and each matrix element DT_{ij} contains the number of times a word in twitterer$_i$ tweets has been assigned to topic t_j and can indicate the probability of his interest in that topic. Therefore, each row represents the probability distribution of the twitterer's interest over the T topics.

$$P_t\big(Flr_i,\ Fnd_j\big) = \frac{TsFnd_j}{TsFndsFlr_i} * TopSim\big(Flr_i,\ Fnd_j\big), \tag{7}$$

where

$$TopicSim\big(Flr_i,\ Fnd_j\big) = 1 - \big|DT_{it} - DT_{jt}\big|$$

and P_t is the transition probability matrix, Fnd_j is friend *j,* Flr_{i} is follower *i*; the twitterer whose updates are being followed is called a *friend,* and the one who is following is called follower. $TsFnd_j$ represents tweets of friend *j,* $TsFndsFlr_i$ denotes tweets sent by friends of follower *i,* and $TopSim(Flr_i,Fnd_j)$ is follower *i* and friend *j* topic similarity. □

The elements of the transition probability matrix are calculated using Eq. 7. The higher the transition probability, the higher the influence from the *j* friend on the follower *i*'s friend in topic *t*. P_t is utilized in getting the topic-specific TwitterRank of the twitterers in a certain topic. An aggregation of the TwitterRank can also be obtained to measure twitterers' overall influence.

Another method to discover influential twitterers in certain topics is through considering the In-degree which can be indicated by the number of followers the twitterers have. The In-degree is currently deployed in Twitter, and it is also utilized in twitterholic.*com* and *wefollow.com* [17].

In [17], however, it is argued that if twitterer *TRa* decides to follow twitterer *TRb,* then this decision is not necessarily based on *TRb's* expertise in the topic since it can be out of courtesy since *TRb* is a follower of *TRa* which does not reflect that the followed person is a real expert in the topic. This is improved in PageRank by considering the link structure of the whole network. It is a way of ranking the importance of a Web page based on the link structure of the pages leading to that page, but again the topic association between the linked pages is not considered in the ranking. So when this approach is applied in Twitter it ignores the similarity in interests among twitterers. Even with the advanced version of PageRank which is the topic-sensitive PageRank, the same transition probability matrix is used for different topics [17]. In [24, 32], a third method for discovering influential twitterers relies on the ratio of retweet, reply and mention of twitterer's published tweets.

In [8], social crawling is used to identify the topics the users are experts in which are extracted from their structured profiles with topic parsing algorithms that access specific fields or extracted automatically from unstructured text provided in the content of their tweets, status message updates or Facebook pages. Linear SVM is used to identify the main topics or subject area followed by ad hoc entity extractor scaled by *tf-idf* score to identify more specific topics in which the user can be considered expert. Feedback from practical experience can adjust found information when the user mutes a topic, refuses to answer or receives negative feedback from another user. This is followed by a topic strengthening algorithm that strengthens the expertise of the user in a topic if he or she has friends that are also experts in the same topic. This process results in creating two indices, the first is the forward index that associates every userId with a scored list of topics, and a series of further scores about a user's behavior (e.g., responsiveness or answer quality). Then, this index is inverted to get the inverted index that has the topicId associated with a scored list of userIds that have expertise in that topic. It also stores scored lists of userIds for features like answer quality and response time.

In addition to the inverted index, a social graph is developed that indexes users' friendship and affiliation information. The friend groups' information can be imported automatically from social networks or added by the users. This social graph is saved in a fixed width ISAM index sorted by user Id. These indices are used to rout the questions to the potential answerers with the highest experience ranks in the question topic.

This method prioritizes the users not only according to their expertise, but also their connectedness and availability. To decide the level of expertise, the user profiles are analyzed and users with profile topics relevant to the question topics are highly ranked and for location-sensitive queries, users with profile locations close to the event discussed are preferred.

Query-independent criteria are used for potential asker/answerer based on their sense of connection, similarity and the answerer availability where answerers are prioritized by being recently online, usually active at similar day time and receiving no recent request to answer messages.

3 Conclusion

Crowdsourcing has been utilized in many areas like security, computer vision, Internet accessibility, adult content filtering, language translation, monitoring of security cameras, improving Web search, text summarization [26], crime suspects detection [14] and disaster relief [13]. In [13], it is stated that crowdsourcing applications like translation, filtering and categorization if efficiently utilized by governments and NGOs can significantly contribute to future disaster relief efforts and saving lives.

It can be seen from all the applications that utilizing crowdsourcing proved to be an efficient and cheap way of solving problems and accomplishing tasks that need human inelegance and cannot be solved by the computer and very expensive and time-consuming to solve by employing experts. The crowdsourcing process starts by mobilizing the crowd to collaborate in solving the problem in hand either by tangible incentives like money as in AMT or using intangible incentives such as entertainment, attention and recognition. Then, the information is collected and aggregated after applying quality assurance and verification methods to filter out malicious or wrong contributions and exclude unreliable participants who can lower the overall quality of the whole crowdsourcing process.

To avoid having unreliable participants, methods for expert discovery are developed. Having plenty of information embedded in the social media accounts about the users can facilitate expert discovery in advance, prior to task appointment.

This may not be readily available for other crowdsourcing applications like AMT where the information about workers is limited and can only be built overtime by experiencing their work performance. Different methods are developed for each step in the crowdsourcing process and choosing the best method can highly rely on the required task and available resources.

References

1. Aaron S, John H, Daniel C (2011) Designing incentives for inexpert human raters. In: Proceedings of the ACM conference on computer supported cooperative work, CSCW'11
2. Aditya P, Scott C (2011) Identifying topical authorities in microblogs. In Proceedings of ACM conference on web search and data mining (WSDM), pp 45–54. https://doi.org/10.1145/1935826.1935843
3. Alexander S, David F (2008) Utility data annotation with amazon mechanical turk. In: First IEEE workshop on internet vision at CVPR'08
4. Bernardo AH, Daniel MR, Fang W (2009) Crowdsourcing, attention and productivity. J Inf Sci 35:758–765. https://doi.org/10.1177/0165551509346786
5. Bin Y, Yan W, Ling L (2015) CrowdTrust: a context-aware trust model for workers selection in crowdsourcing environments. In: 22nd IEEE international conference on web services (IEEE ICWS, research track, acceptance rate 17.4%), June 27–July 2, 2015, New York, USA
6. Catherine G, Matthew L (2010) Crowdsourcing document relevance assessment with mechanical turk. In: Proceedings of the NAACL HLT 2010 workshop on creating speech and language data with Amazon's mechanical turk, pp 172–179
7. Charles LAC, Gordon VC, Elizabeth AT (2000) Relevance ranking for one to three term queries. Inf Process Manag 36(2):291–311
8. Damon H, Sepandar DK (2010) The anatomy of a large-scale social search engine. In: Proceedings of 19th ACM international conference on world wide web. ACM, New York, pp 431–440. https://doi.org/10.1145/1772690.1772735
9. Gabriella K, Jaap K, Natasa M (2013) An analysis of human factors and label accuracy in crowdsourcing relevance judgments. Inf Retr 16(2):138–178. https://doi.org/10.1007/s10791-012-9205-0
10. Galen P, Iyad R, Wei P et al (2011) Time-critical social mobilization. Science 334:509–512. https://doi.org/10.1126/science.1205869
11. Gianluca S, Gang W, Manuel E et al (2013) Follow the green: growth and dynamics in twitter follower markets. In: Proceedings of IMC
12. Haoqi Z, Eric H, Yiling C et al (2012) Task routing for prediction tasks. In: Proceeding of 11th international conference autonomous agents and multiagent systems, vol 2. International foundation for autonomous agents and multi-agent Systems, Richland, pp 889–896
13. Huiji G, Geoffrey B, Goolsby Rebecca (2011) Harnessing the crowdsourcing power of social media for disaster relief. Intell Syst IEEE 26:10–14
14. Iyad R, Sohan D, Alex R et al (2013) Global manhunt pushes the limits of social mobilization. Computer 46:68–75. https://doi.org/10.1109/mc.2012.295
15. Jacob W, Paul R, Ting-fan W et al (2009) Whose vote should count more: optimal integration of labels from labelers of unknown expertise. In: Neural information processing systems (NIPS)'09, pp 2035–2043
16. Jia D, Wei D, Richard S et al (2009) ImageNet: a large-scale hierarchical image database. In: CVPR
17. Jianshu W, Ee Peng L, Jing J et al (2010) Twitterrank: finding topic-sensitive influential twitterers. In: Proceedings of ACM conference on web search and data mining (WSDM)
18. Jing W, Panagiotis GI, Foster P (2016) Cost-effective quality assurance in crowd labeling. Inf Syst Res (**Forthcoming**), Dec 2016, NYU Working Paper No. 2451/31833
19. Ju R, Yaoxue Z, Kuan Z et al (2015) SACRM: social aware crowdsourcing with reputation management in mobile sensing. Comput Commun (Elsevier) 65(1):55–65
20. Julien B, Shourya R, Gueyoung J et al (2013) Crowdsourcing translation by leveraging tournament selection and lattice-based string alignment. In: AAAI conference on human computation and crowdsourcing (HCOMP) (Works in Progress/Demos)

21. Ke M, Licia C, Mark H et al (2015) Survey of the use of crowdsourcing in software engineering. RN 15 (2015), 01
22. Kyumin L, Prithivi T, James C (2013). Crowdturfers, campaigns, and social media: tracking and revealing crowdsourced manipulation of social media. In: Proceedings of ICWSM
23. Kyumin L, Steve W, Hancheng G (2014) Characterizing and automatically detecting crowdturfing in Fiverr and Twitter. Soc Netw Anal Min 5(1):1–16
24. Leavitt A, Evan B, David F et al (2009) The influentials: new approaches for analyzing influence on twitter. Web Ecol Proj 4:1–18
25. Liang W, Huan L (2017) Detecting crowdturfing in social media. Encyclopedia of social network analysis and mining. Springer, New York, pp 1–9. https://doi.org/10.1007/978-1-4614-7163-9_110196-1
26. Luis VA (2006) Games with a purpose. Computer 39:92–94. https://doi.org/10.1109/mc.2006.196
27. Luis VA, Laura D (2004) Labeling images with a computer game. In: SIGCHI conference on Human factors in computing systems, pp 319–326
28. Manuel C, Lorenzo C, Andrea VA et al (2012) Finding red balloons with "split" contracts: robustness to individuals' selfishness. In: ACM symposium on theory of computing (STOC)
29. Manuel C, Iyad R, Victoriano I et al (2016) Searching for someone. Illustrated by Beatriz Travieso. Published in MIT media lab. Sponsored by the Data61 Unit at CSIRO. https://medium.com/mit-media-lab/searching-for-someone-688f6c12ff42#.tlleaq622
30. Maribel A, Amrapali Z, Elena S et al (2013) Crowdsourcing linked data quality assessment. In: Harith A et al (eds) ISWC 2013, Part II. LNCS, vol 8219. Springer, Heidelberg, pp 260–276
31. Marti L, Stefan V (2011) Dirty jobs: the role of freelance labor in web service abuse. In: Proceedings of the 20th USENIX security symposium, USESEC'11, San Francisco, CA
32. Meeyoung C, Hamed H, Fabriıcio B et al (2010) Measuring user influence in twitter: the million follower fallacy. In: Proceedings of AAAI conference on weblogs and social media (ICWSM)
33. Mohammad A, Boualem B, Aleksandar I et al (2013) Quality control in crowdsourcing systems: issues and directions. IEEE Internet Comput 17(2):76–81. https://doi.org/10.1109/MIC.2013.20
34. Nguyen Q, Nguyen T, Lam T et al (2013) An evaluation of aggregation techniques in crowdsourcing. WISE 2:1–15
35. Panagiotis I, Foster P, Jing W (2010) Quality management on amazon mechanical turk. In Proceedings of the ACM SIGKDD workshop on human computation (HCOMP'10), pp 64–67
36. Pei-Yun H, Prem M, Vikas S (2009) Data quality from crowdsourcing: a study of annotation selection criteria. In: Proceedings of the NAACL HLT workshop on active learning for natural language processing. Association for Computational Linguistics, pp 27–35
37. Peter W, Pietro P (2010) Online crowdsourcing: rating annotators and obtaining cost effective labels. In: IEEE conference on computer vision and pattern recognition workshops (ACVHL)
38. Peter W, Steve B, Serge B et al (2010) The multidimensional wisdom of crowds. In: Neural information processing systems conference (NIPS), vol 6
39. Petros V, Hector G-M, Kerui H et al (2012) Max algorithms in crowdsourcing environments. In: Proceedings of the 2012 international conference on the world wide web, 2012, pp 989–998. http://dx.doi.org/10.1145/2187836.2187969
40. Philip D, Skene AM (1979) Maximum likelihood estimation of observer error-rates using the EM algorithm. J R Stat Soc Ser C (Appl Stat) 28(1):20–28
41. Rion S, Brendan O, Daniel J et al (2008) Cheap and fast—but is it good? Evaluating non-expert annotations for natural language tasks. In: EMNLP
42. Sam M, Mike JJ, Didier GL (2014) A flexible framework for assessing the quality of crowdsourced data. In: 17th annual international AGILE conference, Castellón, Spain
43. Saptarshi G, Naveen S, Fabricio B et al (2012) Cognos: crowdsourcing search for topic experts in microblogs. In: Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval, pp 575–590
44. Shih-Wen H, Wai-Tat F (2013) Enhancing reliability using peer consistency evaluation in human computation. In: Computer supported cooperative work (CSCW), San Antonio, TX, USA, pp 639–648. https://doi.org/10.1145/2441776.2441847
45. Stefanie N, Stefan R (2010) How reliable are annotations via crowdsourcing? A study about inter-annotator agreement for multi-label image annotation. In: The 11th ACM international conference on multimedia information retrieval (MIR), Philadelphia, USA, pp 29–31
46. Twitter: Who to Follow. http://twitter.com/#!/who_to_follow
47. Victor N, Iyad R, Manuel C et al (2012) Verification in Referral-Based Crowdsourcing. PLOS One 7(10):e45924
48. Xiaohang Z, Guoliang L, Jianhua F (2016) Crowdsourced top-k algorithms: an experimental evaluation. PVLDB 9(8):612–623

49. Yu-An S, Shourya R, Greg DL (2011) Beyond independent agreement: a tournament selection approach for quality assurance of human computation tasks. In: Proceedings of HCOMP11: the 3rd workshop on human computation

Lobna Nassar was granted the B.Sc. degree and awarded the Graduate Diploma both in Computer Science from the American University in Cairo (AUC) in 1996 and 2004, respectively. She worked as a teaching and laboratory assistant at the Computer Science Department at AUC and was awarded the Certificate of Academic Honor for Outstanding Achievement. She was awarded the M.Sc. in Information Technology-Knowledge and data management (with Distinction) from the British University in Dubai (BUiD) in association with the University of Edinburg in 2011. She received her Ph.D. in Electrical and Computer Engineering from the University of Waterloo (UOW), ON. Canada, in December 2015. Her research is centered on utilizing Information Retrieval (IR) techniques in developing context-aware systems that enhance information dissemination in Vehicular Ad Hoc Networks (VANETs). Areas of interest include Information Retrieval, Crowdsourcing, Human–Computer Interaction, Artificial Intelligence, Knowledge Representation and Knowledge Engineering.

Fakhri Karray is UW Research Chair Professor in Electrical and Computer Engineering and co-Director of the Pattern Analysis and Machine Intelligence Center. He received his Ph.D. from the University of Illinois, USA (1989), in the area of systems and control. Dr. Karray's research interests are in the areas of intelligent mechatronics and transportation systems, soft computing, autonomous machines and natural man–machine interaction. He is the author of more than 300 technical articles, 14 US patents, a major textbook on soft computing and more than 20 textbook chapters. He has chaired/co-chaired more than 15 international conferences. He has also served as the associate editor/guest editor for more than 10 journals, including the IEEE Transactions on Systems Man Cybernetics (B), the IEEE Transactions on Neural Networks and Learning, the IEEE Transactions on Mechatronics, the IEEE Computational Intelligence Magazine. He is the Chair of the IEEE Computational Intelligence Society Chapter in Waterloo, Canada.