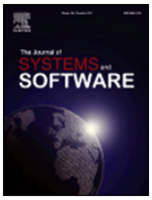
Contents lists available at ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss



General framework, opportunities and challenges for crowdsourcing techniques: A Comprehensive survey

Shahzad Sarwar Bhatti, Xiaofeng Gao*, Guihai Chen

The Shanghai Key Laboratory of Scalable Computing and Systems, Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, P.R. China

ARTICLE INFO

Article history:
Received 20 August 2018
Revised 1 March 2020
Accepted 19 April 2020
Available online 29 April 2020

Keywords:
Crowdsourcing
Framework
Incentives
Decomposition
Aggregation
Reputation

ABSTRACT

Crowdsourcing, a distributed human problem-solving paradigm is an active research area which has attracted significant attention in the fields of computer science, business, and information systems. Crowdsourcing holds novelty with advantages like open innovation, scalability, and cost-efficiency. Although considerable research work is performed, however, a survey on the crowdsourcing process-technology has not been divulged yet. In this paper, we present a systematic survey of crowdsourcing in focussing emerging techniques and approaches for improving conventional and developing future crowdsourcing systems. We first present a simplified definition of crowdsourcing. Then, we propose a framework based on three major components, synthesize a wide spectrum of existing studies for various dimensions of the framework. According to the framework, we first introduce the initialization step, including task design, task settings, and incentive mechanisms. Next, in the implementation step, we look into task decomposition, crowd and platform selection, and task assignment. In the last step, we discuss different answer aggregation techniques, validation methods and reward tactics, and reputation management. Finally, we identify open issues and suggest possible research directions for the future.

1. Introduction

Crowdsourcing (CS), is an emerging paradigm that has been receiving great attraction since 2006. It is a method of solving a specific set of functions by outsourcing and utilizing distributed human computational capabilities through the Internet. Crowdsourcing reveals new approaches to harvest distributed human intelligence and provides unprecedented opportunities to people in sharing their observations and knowledge with the rest of the world Brabham (2008). This form of harvesting human wisdom has proven to be a promising problem-solving and business production model Allahbakhsh et al. (2013). The web-based model is capable of leveraging ingenuity, aggregating talent, collaborative intelligence, and reducing cost and processing time. Now, it's widespread in supporting various applications and has achieved remarkable feats in the image or video classification or labeling Krizhevsky et al. (2012), natural language processing Inel et al. (2014), software development Design and Build High-Quality Software with Crowdsourcing (0000), character recognition von Ahn et al. (2008), designing fabrics Threadless (0000), and many other applications. Different areas making use of crowdsourcing are shown in Fig. 1.

Crowdsourcing is a neologism which refers to the compound fusion of three key defining elements, i.e., 'crowd', 'outsourcing' and 'social web' Saxton et al. (2013) as shown in Fig. 2. The success of outsourcing and advanced Internet technologies are significant factors for the attention to crowdsourcing. The evolution of Internet technology and its ubiquitous access gives a new dimension to this phenomenon in providing user interactivity and bringing massive intelligence to solve various problems at an affordable price. The tremendous growth of crowdsourcing is its inherited power of parallel processing, i.e., it enables to perform multiple tasks simultaneously, reducing the time as well. For example, several people working simultaneously on labeling images reduce the overall time required. Another reason is that some tasks that are hard for machines or individuals like audio translation and image tagging can be easily performed using crowdsourcing. People are always ready to do tasks for micro-payments, and in some instances, no payment which decreases overall expenses in generating quality work.

The notion of crowdsourcing is by no means new; the concept was developed in the 18^{th} century and has been used for many years. In 1714, the British government required to develop a solution and offered £ 20,000 through an open call for so-called *The Longitude Problem* Sobel (2007), as this made sailing dangerous,

* Corresponding author.
E-mail addresses: shahzadbhatti@sjtu.edu.cn (S.S. Bhatti), gao-xf@cs.sjtu.edu.cn, gaoxiaofeng@sjtu.edu.cn (X. Gao), gchen@cs.sjtu.edu.cn (G. Chen).

https://doi.org/10.1016/j.jss.2020.110611
0164-1212/

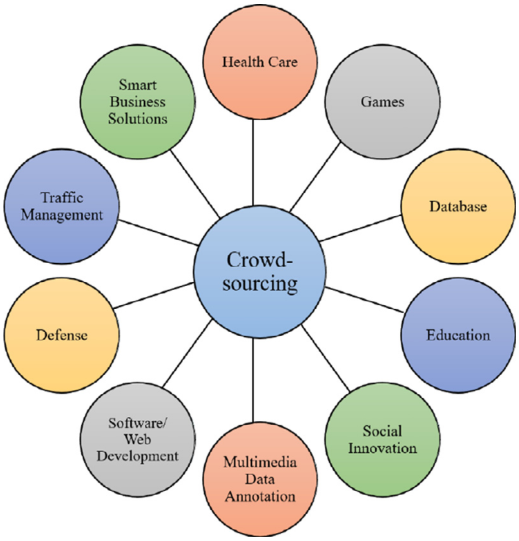


Fig. 1. Areas leveraging crowdsourcing.

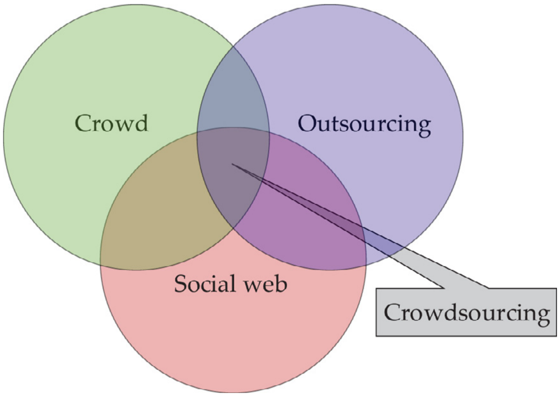


Fig. 2. Three key defining elements for crowdsourcing.

killing a large number of sailors or lost on unknown islands due to absence of navigational parameters. This first-ever event of crowdsourcing, in which the winner was John Harrison, a clockmaker and a self-taught English carpenter Chrum (0000). In 1810, when Napoleon expanded his European empire, a large number of soldiers were employed in the armies, so it was required to preserve the food. The French government announced and offered 12,000 francs to the person who invented a practical method to store and avoid wastage of food. A prize was awarded for designing canned food Stol et al. (2017).

In 1884, crowd corrected and updated catalog of the *Oxford English Dictionary* Lynch (0000). The famous Japanese motor company arranged a public competition of logo design in 1936. The new logo 'Toyota' was chosen from 27,000 submissions received from the crowd From “TOYODA” to “TOYOTA” (0000). In 1955, the Australian government held a contest to design a building for Sydney harbor. The winner among 233 contestants got the prize of £ 5,000. The winning design of Sydney's Opera house is one of state of the art crowdsourced architectural designs Stol et al. (2017). During early 1998, Eli Lilly, the American multinational pharmaceutical company, created an online platform ‘InnoCentive’ that deals with business intelligence and embrace the power of the crowd InnoCentive (0000). These are few earlier crowd related examples from the past.

The concept of crowdsourcing expanded rapidly and continued to take hold in the 21st century. In 2003, the online encyclopedia *Wikipedia* was one of the best illustrations in acquiring collective wisdom and knowledge through crowdsourcing. The online video community *YouTube* initiated in 2005, is an example of crowdsourcing entertainment. Numerous other companies registered in Fortune 500 Fortune Data Store (0000) are dependent on crowdsourcing to solve different tasks. In 2009, NESTA in the UK announced three winners among 355 groups in the Big Green Challenge for reducing CO_2 emissions, and each received a prize of GBP 300,000 Marjanovic et al. (2012). Above examples exhibit the power and revolutionary nature of the crowd in performing different tasks.

Since crowdsourcing has been an active research area in the past few years, there have been a variety of surveys discussing this area. One kind of surveys is concentrated, as mainly focus on the single aspect of crowdsourcing such as incentive engineering techniques in crowdsourcing Muldoon et al. (2018), privacy-preserving issues in crowdsourcing Alkharashi and Renauld (2018), quality control in crowdsourcing Jin et al. (2018); Daniel et al. (2018), applications of crowdsourcing in software engineering Mao et al. (2017); Sar et al. (2019), medical image analysis Orting et al. (2019) and data analytics Li et al. (2016), or statistical analysis of crowdsourcing research Tarrell et al. (2013); Aris (2017).

Another kind of surveys has attempted to comprehensively discuss crowdsourcing in different aspects Yuen et al. (2011); Hetmank (2013); Zhao and Zhu (2014); Chittilappilly et al. (2016); Ghezzi et al. (2017); Nassar and Karray (2019). Yuen et al. Yuen et al. (2011) presented a literature survey on different aspects of crowdsourcing systems. In addition to a taxonomy of CS, they categorized the studies in terms of the applications (voting systems, information sharing system, games, and creative systems), algorithms, performances (user participation, quality management, and cheating detection), and available datasets. Hetmank Hetmank (2013) focuses on system architecture design useful for supporting the process of crowdsourcing. Through a systematic literature review, the author identified components and functions that are implemented in traditional CS systems. Four components along with their functions were presented including user management (registration, evaluation, group formation and coordination mechanisms), task management (design and assign task), contribution management (evaluate and select contribution), and workflow management (define and manage workflow for a task). Zhao et al. Zhao and Zhu (2014) and Ghezzi et al. Ghezzi et al. (2017) summarized the current status and future avenues of research in crowdsourcing from a global view of information and systems; however, technical details are missed in these studies. While Ghezzi et al. (2017) introduced the framework for the CS process, where process-technology details are missed in this work. Chittilappilly et al. Chittilappilly et al. (2016) provided a brief description and limitations of existing technologies for solving various CS problems. They review existing studies in areas of workers' motivation and engagement, task allocation, and quality control mechanisms. They also discussed the implementation of different CS techniques on a spatial crowdsourcing platform, gMission. Nassar et al. Nassar and Karray (2019) introduced a crowdsourcing process. The process mainly contains five modules: incentives, quality control approaches, collection and verification methods, aggregation, and topical experts discovery. They review different methods used to accomplish each processing step.

Taking into consideration the motivation for this research is that, the existing studies by and large ignore the crowdsourcing processes from the technical perspective that is typical of CS endeavours, failing to provide a systematic framework in the design and implementation of CS systems. Therefore, we present a survey of crowdsourcing process-technology in focusing on emerging techniques and approaches for developing and improving the pro-

Table 1
The comparison of existing surveys and reviews on crowdsourcing.

Type	Ref.	Advances	Differences
Concentrated	Muldoon et al.Muldoon et al. (2018)	Survey on **incentive** engineering mechanisms	Consider more about game theoretic approaches
	Alkharashi et al.Alkharashi and Renauld (2018)	Overview of **privacy** protection aspects	Discuss categories, principles, concerns and enhancement
	Mao et al.Mao et al. (2017)	Survey on **application** in software engg.	Analyze various domains, tasks, platforms and applications
	Sari et al.Sar et al. (2019)	Systematic review of **application** in software engg.	Discuss business models, tools, platforms, processes and economics
	Orting et al.Orting et al. (2019)	Survey on **application** in medical image analysis	Analyze applications, interaction, platform, evaluation and results
	Li et al.Li et al. (2016)	Survey on **application** in data analytics	Consider quality control, cost control and latency control
	Jin et al.Jin et al. (2018)	Survey on **quality** control	Integrate design methods and statistical modelling methods
	Daniel et al.Daniel et al. (2018)	Survey on **quality** control	Consider models, assessment and assurance
	Tarrell et al.Tarrell et al. (2013)	Summarize **statistical analysis**	Include foci, framework and keyword analysis
	ArisAris (2017)	Review **statistical analysis and taxonomy**	Include classification and respective themes
Comprehensive	Yuen et al.Yuen et al. (2011)	Survey on CS systems	Only consider applications, algorithms, performance and datasets
	HetmankHetmank (2013)	Review components and functions of CS systems	Include fewer implemented components
	Zhao et al.Zhao and Zhu (2014)	Summarize current status and future directions	Less technical details
	Chittilappilly et al.Chittilappilly et al. (2016)	Survey of different techniques	only discuss incentive, assignment and quality control techniques
	Ghezzi et al.Ghezzi et al. (2017)	Review current state and future directions	Adopt a framework with fewer components and less details
	Nassar et al.Nassar and Karray (2019)	Overview of CS process	Adopt a framework with fewer components and less details

cess of CS systems. We propose a framework of the crowdsourcing process, synthesize a wide spectrum of existing studies for various dimensions of the framework. The paper is oriented to the emerging techniques and technical methodologies provided for different components and functions of the framework. The work can be seen as an extension of Hetmank (2013) with plenty of full advances during recent years. The further survey on the related works is justifiable and meaningful in our paper. A summary of advances and differences of existing survey papers is outlined in Table 1.

Specifically, the key contributions of our paper are summarized as follows.

- We analyze existing important definitions of crowdsourcing and present a simple definition.
- We review the existing surveys and summarize the advances and differences in Table 1.
- We propose a systematic and modifiable framework which provides the complete workflow of the crowdsourcing process.
- We synthesize a broad spectrum of existing studies and map them through the lenses of our framework.
- We present a contemporary analysis of emerging techniques in improving conventional and developing CS systems.
- Based on the analysis, we identify limitations, particular challenges and future research directions.

The survey is organized as follows. First, preliminaries and overview of the framework along with workflow are provided (Section 2). The later sections are structured according to corresponding parts of the framework. The nine dimensions of the framework according to which we analyzed the papers are described in more details (Sections 3 to 11). Several significant challenges are discussed and future emanating research directions are outlined (Section 12). In the end, a brief conclusion of the survey is presented (Section 13).

2. Preliminaries and framework

We first present research methodology, review a few definitions, typical system model and then refine our focus to the framework of crowdsourcing process.

2.1. Research methodology

To conduct a comprehensive survey of all publications related to the crowdsourcing process, it is necessary to do a careful and rigorous search. The search process includes a paper selection from the existing literature and then find the articles relevant to different segments of the framework of the crowdsourcing process. We queried contributions from 2010 onward for manageable and up-to-date references.

We used the following well-presentative keywords in either the title, abstract, or keywords: “crowdsourc”, “crowd-sourc”, and “crowd sourc”. The other domain-specific terms were “define or defining or definition”, “framework”, “task design”, “task setting”, “incentive or motivation”, “decomposition or fragment or split”, “platform”, “assignment or allocation”, “aggregation or integration or integrate” and “reputation or evaluation”. Consequently, articles were retrieved from popular search engines and resources, including *Elsevier, dblp, ACM Digital Library, Science Direct, IEEE Xplore, Springer*, and *Google Scholar*. We also made manual retrievals of some major journals and conference proceedings to find the relevant literature. The articles published in English and available in full-text in journal, conference or workshop were selected. The criterion of whether to include a paper or not in our survey is that the paper contains the word “crowdsourcing” in the title, abstract, keywords, or description on crowdsourcing relevant for this survey.

The search identified 1179 papers. Further, we performed a manual search and finally have a total of 234 articles for this study. We also considered additional papers from prior knowledge. To avoid missing any relevant paper, we checked the reference section of selected papers in the final pool and searched the papers in other databases. Moreover, to the best of our knowledge, the proposed research framework of the crowdsourcing process focusing techniques and methods has not been explored through applied research.

Representative journals are Elsevier JSS, IJIM, MCM, ACM CSUR, ACM TWEB, Commun. ACM, ACM Crossroads, IEEE/ACM TON, VLDB, IEEE TKDE, IEEE Internet computing, IEEE Software, IEEE Access, IEEE JSAC, IEEE TIFS, IEEE TVCG, IEEE TSP, IEEE ITPro, IJHCI, IJMR, IJCA, JIS, iCS, IJWIS, Multimedia Syst., Comput. J., JHRI, CoRR, Eng. Appl. of AI, JCIS, IJAET, Business Ethics: A European Review, BJELL, and ISR. Whereas representative conferences are CHI, ISCW, CSCW, AMCIS, ICOCI, Social Com., ISC, CTS, SMC, HICSS, MMSP, SOSE, MindTrek, SIGMOD, CIDR, AAAI, CSCWD, PerCom, SOCA, IPDPS, AVI, WSDM, EDBT, RTEICT, WWW, MobiSys, UIST, CLOUD, ICC, CSE, PRIMA, ACL, MIPaC, ICDCS, KDD, MOD, UbiCrowd, ICDE, NIPS, HCOMP, IWQoS, CTS, ECDC, KSEM, ICML, ICA3PP, WIMS, IMIS, ICEEE.

2.2. Crowdsourcing definition

Crowdsourcing has enjoyed a recent rise in popularity and gained much attraction in scholarly research, eminent practitioners, and industrial communities. Two journalists Jeff Howe and Mark Robinson coined the modern term ‘Crowdsourcing’, which first appeared in the article ‘The Rise of Crowdsourcing’ in the June 2006 issue of Wired magazine Howe (b). Howe offered the following definition:

“Simply defined, crowdsourcing represents the act of a company or institution taking a function once performed by employees and outsourcing it to an undefined (and generally large) network of people in the form of an open call.” Howe (a).

Researchers and practitioners have been involved since then, offered many definitions with subtle differences Brabham (2009); Howe (2008). To define the crowdsourcing concept, few researchers defined crowdsourcing as a problem solving approach Doan et al. (2011); Afuah and Tucci (2012). While in some other cases, it was defined according to applied contexts such as small and medium enterprises Maiolini (2011) and business-to-business (B2B) application Kärkkäinen et al. (2012). Moreover, some researchers compared crowdsourcing with outsourcing Saxton et al. (2013). Estellés and González Estellés-Arolas and González-Ladrón-De-Guevara (2012) synthesized existing literature and provided an integrated and global definition that considers several aspects of initiator, crowd, and process.

There is a relatively high number of idiosyncratic definitions and perceptions of crowdsourcing in the literature Wazny (2017). Many researchers attempted to propose their definitions based on different theoretical models and diverse practices. In Estellés-Arolas and González-Ladrón-De-Guevara (2012), about 40 definitions were collected and analyzed to extract common elements and establish essential characteristics. Around 17 definitions were found and categorized in different groups Hetmank (2013). Recently, several existing definitions of crowdsourcing are analyzed Bozat (2017). The definition became blurry and ambiguous when some researchers even stated multiple versions of the definition such as Vukovic Vuković (2009); Vukovic and Bartolini (2010), BrabhamBrabham (2008, 2010) and Whitla Whitla (2009). Some of the definitions contradict each other and causes misunderstandings Hopkins (2011); Hosseini et al. (2015). As the analysis progresses, it has been shown that distinct definitions of crowdsourcing exist; however, with a lack of consensus and a certain semantic confusion.

A firm and commonly accepted definition of crowdsourcing by integrating a variety of interpretations might be infeasible and has yet to emerge. We agree with this updated definition in Estellés-Arolas and González-Ladrón-De-Guevara (2012) but noted to be wordy, complicated and less wieldy in practice Brabham (2013a); Thuan and Antunes (2013); Thuan et al. (2016). It does, however, hint at various research foci involved in developing better understanding and nuances of this phenomenon. Addressing the confusion of academic and popular discussion of crowdsourcing, in this paper, as a first contribution, following we propose a workable and straightforward definition of crowdsourcing that differentiates it from e-commerce, e-business, Web 2.0, outsourcing, and other relevant concepts.

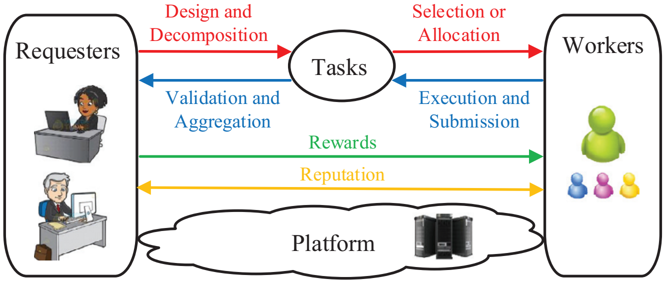


Fig. 3. Components and processes in CS systems.

“Crowdsourcing is an online distributed problem-solving paradigm, in which an individual, company, or organization publishes defined task(s) to the dynamic crowd through a flexible open call to leverage human intelligence, knowledge, skill, work and experience”.

The topic has been discussed within the research community, press, books and academia Hossain and Kauranen (2015); Mihalcea and Chklovski (2004); Weld et al. (2008). As is typical for an emerging area, the term has appeared under several naming taxonomies such as social system, collaborative system, peer production, community system, user-powered system, social networking, social search, crowd wisdom, collective intelligence, smart mobs, user-generated content, human collaboration, human computation and mass collaboration. These alternatives are expected, given a nascent state of the idea and applications. We anchor on ‘crowdsourcing’, as the term dominates the literature.

2.3. System model

The key elements and various processes in crowdsourcing are shown in Fig. 3. In general, each CS system includes the following four key elements: (1) *task*, which is the outsourced object with a set of instructions, having various characteristics among which task design and rewards are typically important; (2) *requester*, also known as crowdsourcer, publisher or seeker, can be an individual, company, or profit/non-profit organization, who publishes tasks to find a solution to the problem; (3) *worker*, also known as a solver, participant, or provider, who possess knowledge, skills, abilities and can provide computational power for accomplishing different types of outsourced task problems. (4) *platform*, which is the mediator between requester and worker, provides efficient measures to organize and manage the entire crowdsourcing process and may undertake some affairs related to requesters.

The general system flow of a crowdsourcing application is as follows. A requester creates tasks with explicit details and forwards tasks to the platform (Step #1). Once tasks arrive at the platform, these are published at the platform. To perform tasks, workers search or browse these tasks in the system or platform can also assign tasks to suitable available workers (Step #2). Workers execute tasks and submit solutions to the requester through the platform (Step #3). The platform forwards these solutions to the requester for validation (Step #4). Finally, workers are rewarded through the platform after requesters verify the submitted solutions. Various examples in research and commercially used platforms are Ama-

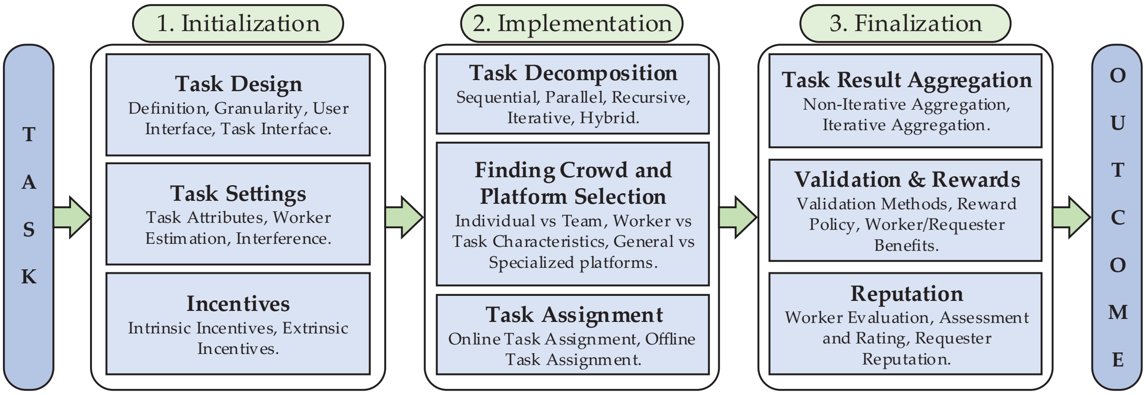

1. Initialization
2. Implementation
3. Finalization
TASK
Task Design
Definition, Granularity, User Interface, Task Interface.
Task Settings
Task Attributes, Worker Estimation, Interference.
Incentives
Intrinsic Incentives, Extrinsic Incentives.
Task Decomposition
Sequential, Parallel, Recursive, Iterative, Hybrid.
Finding Crowd and Platform Selection
Individual vs Team, Worker vs Task Characteristics, General vs Specialized platforms.
Task Assignment
Online Task Assignment, Offline Task Assignment.
Task Result Aggregation
Non-Iterative Aggregation, Iterative Aggregation.
Validation & Rewards
Validation Methods, Reward Policy, Worker/Requester Benefits.
Reputation
Worker Evaluation, Assessment and Rating, Requester Reputation.
OUTCOME

Fig. 4. A research framework of the crowdsourcing process.

zon Mechanical Turk (AMT or MTurk) Amazon Mechanical Turk (0000), FigureEight FigureEight (0000) (formerly known as CrowdFlower), InnoCentive InnoCentive (0000), UpWork Upwork (0000), and others.

2.4. Overview of framework

Crowdsourcing process involves different steps in performing tasks from the beginning until the completion of a task. Various workflows and frameworks discussed the process in the literature are simple, with a limited number of processes Liu and Lu (2016); Silva et al. (2017); Ghezzi et al. (2017); Chittilappilly et al. (2016); Tarrell et al. (2013). Due to emerging and relative novelty, the contours of a unified framework are not available. Inline with the evolution of crowdsourcing applications, the number of processes in crowdsourcing has also increased. Towards an attempt, to devise the multifaceted contributions, we propose a systematic and modifiable research framework of the crowdsourcing process with various segments and associated sub-segments. We divided our framework mainly in three steps i.e., *initialization*, *implementation*, and *finalization* as shown in Fig. 4.

The first step is the initialization, i.e., task preparation. A requester should design task properly as crowd contribution can be increased and cheat submissions can be avoided through appropriate designs. The process is comprised of task design, task settings, and incentives. In task design, a requester describes the task and designs a user-friendly interface for different task types. The interface must be provided with a way for the worker to view the relevant information and to provide answers quickly. A task has a set of attributes for a thorough understanding. The requester must take care to avoid inter-task interference such that the execution of any task or sub-task is not affecting other tasks. The requester estimates the workforce required to perform the task. Compensation is vital to attract the workers as workers perform tasks mainly based on offered incentives.

The second step is the implementation of a task, in which the requester publishes tasks and workers perform these tasks. Before forwarding a task to workers, a requester can decompose the task if it is complex and decomposable. The requester should split the task into subtasks and avoid inter-subtask interference in a way that the entire task could be performed easily. Afterwards, requester searches for appropriate workers and selects a suitable platform to publish tasks. The tasks are allocated to the workers based on individual or team-oriented, worker characteristics, or task requirement. To find the workers and a suitable platform, a requester can select a platform from the existing crowdsourcing market such as AMT Amazon Mechanical Turk (0000), FigureEight FigureEight (0000). Crowd selection is essential, as the assignment of the right task to the right worker is a critical factor for the success of crowdsourcing. The workers are available online or offline, and based on their availability, task assignment is performed online or offline accordingly.

In the final step, various activities are accomplished to get the final output. The workers' submissions received from the previous step are aggregated to find the right answer, which is the final solution. Upon validation of results from the requester, workers are rewarded in the form of intrinsic or extrinsic incentives. The outcome of the validation is used to evaluate user reputation, which further guides about task allocations in future.

Depending on the application and task characteristics, the segments or sub-segments in the proposed framework can be rearranged, customized or executed iteratively. Researchers and practitioners can design and improve the process of their systems using the proposed framework and customize their systems according to their requirements. For instance, Amsterdamer et al. (2013); Sun et al. (2014) used the customized processes; the output is the integrated answers which are further used as the input during the next iteration. Now, we describe steps involved in the framework in chronological order.

3. Task design

In the proposed framework, the first step is initialization, and task design is the first activity in the crowdsourcing process. Task design is an important aspect of crowdsourcing. It is a model consisting of several components using which the requester explains a task to workers through semantic and visual presentations. The design of a task is central to the success of the CS system. Efforts should be made to design a task simple and unambiguous Gurari et al. (2016). There is no one-size-fits-all solution for designing a task. In general, the approach is more important that requesters select in designing a task such as fixed templates, customized, manual or wizard, and workflows. The design features and clarity of instructions affect the performance of workers in performing tasks. A requester should optimize the task design through tradeoffs among quality, rate and cost of work. Well-designed tasks have a significant effect on workers' ability in performing tasks quickly and accurately, such as data collection task, an innovation model, or a fundraising scheme. To illustrate task design, we identify five important factors: *task definition, task granularity, task complexity, task types, task duration* and *task interface*. A taxonomy of task design is shown in Fig. 5.

3.1. Task definition

Task definition is the description of the task the requester provides to workers for adequate performance in crowdsourcing. Major components are information and keywords explaining the na-

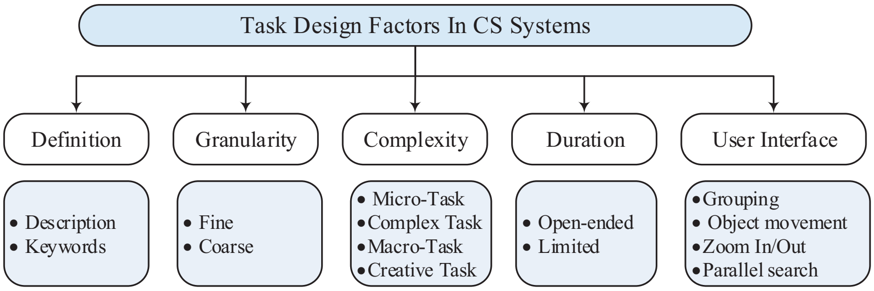


Fig. 5. Task design factors in crowdsourcing.

ture of the task, and timelines, and other components are requirements and constraints for accomplishing the task Aris and Din (2016). A requester can specify the geographical location of workers, mention the required qualification, or acceptance criteria. For instance, in AMT Amazon Mechanical Turk (0000), requesters can specify the percentage of accepted works of workers or describe that workers from only a specific country can participate in performing a particular task. With the help of this set, requesters can specify the eligibility criteria to evaluate workers before accepting their contributions Allahbakhsh et al. (2013). Based on the presentation of tasks to workers, the task definition can be classified into *well-structured*, the solution to the problem is clearly defined with clear boundaries of knowledge domains, and *unstructured*, there is not a defined solution having an ambiguous boundary of knowledge domains Nakatsu et al. (2014).

3.2. Task granularity

Task granularity is one of the main aspects of designing crowdsourcing tasks. It is a way to which tasks can be divided into simple sub-tasks. Based on whether or not a task can be divided into smaller sub-tasks, task granularity can be classified into *fine* and *coarse* levels Allahbakhsh et al. (2015); Aris (2017). The lower the granularity, the higher the precision and vice versa. For example, in the image labeling level of task granularity is fine as each worker performs a relatively small portion of the overall task; whereas, in developing the components of a large software system granularity is coarse Garcia-Molina et al. (2016).

3.3. Task complexity

Task complexity is another useful dimension in designing tasks. It refers to the levels of complexity of the tasks and generally classified as simple or complex level. Existing studies have considered the dimension of task complexity and classified tasks according to their nature. The study in Rouse (2010) categorized the tasks as simple, sophisticated and moderate tasks. Another study Schenk and Guittard (2009) described three different types of tasks such as routine, complex and creative tasks. In Allahbakhsh et al. (2013), tasks are classified into two broad types: simple and complex. Another useful dimension is how a task can be achieved, i.e., individually, competitively, or iteratively Hetmank (2013).

Given that these two dimensions are important in the classification of tasks, we adapted and extended the classification proposed in Schenk and Guittard (2011), who suggested two similar dimensions to classify tasks. In the first dimension, we classify tasks as *micro, complex, creative*, or *macro* task. Micro-tasks can be performed with general skills; complex tasks require knowledge and expertise to solve a problem and usually can be decomposed into smaller sub-tasks; creative tasks depend on individual creativity and uniqueness; macro-tasks are non-decomposable tasks, require expert skills, knowledge and often involve collaboration among workers. It is worth noting that creative tasks‘ objectives usually are to find solutions for particular problems, while most complex tasks may require certain levels of creativity. Consequently, the two categories are not much different for some specific tasks. Often, it is hard to identify precisely the type of task fell along these dimensions.

The other dimension is to distinguish between the *integrative, selective* and *collaboratively* nature of tasks. This distinction refers to the participation mode representing how tasks can be performed individually, competitively, or iteratively. A brief description of the classification of tasks is given below.

3.3.1. Micro-Tasks

The micro-tasks are independent, straightforward, and can be performed in short amounts of time by individual workers. These tasks have fine granularity and do not require cognitive efforts or specific expertise instead accomplish through general skills by ordinary workers. The workers who complete the tasks usually compensated through micro-payments Allahbakhsh et al. (2015). Multiple micro-tasks can be grouped, known as Human Intelligence Task (HIT) Li et al. (2016). Micro-tasks require human intelligence, thus simple for humans but difficult for computers. There are many examples of crowdsourcing for micro-tasks, including but not limited to tagging/labeling images, transcribing audio, classification of galaxies, translating fragments of text, reading Captchas, rating and ranking, verification and validation, and many more. Various CS platforms are oriented to micro-tasks such as *MTurk* Amazon Mechanical Turk (0000), *FigureEight* FigureEight (0000), and *microWorkers* work (0000).

In micro-task crowdsourcing, tasks are generally assigned to more than one worker to improve the performance; each worker independently performs the task. Finally, the requester can select the valid answer from the multiple answers of redundantly assigned individual workers Yuen et al. (2011), or can aggregate the responses of multiple individual workers, e.g., aggregation of labeling work in data mining tasks Xintong et al. (2014), or multiple workers are asked to perform a task collaboratively; for example, workers perform concurrently for data collection tasks Park and Widom (2014).

3.3.2. Complex task

In real crowdsourcing, many tasks require skills, knowledge, and computational efforts, referred to as *complex tasks*. These tasks involve knowledge incentive skills of substantial domains, and cannot be allocated, as a whole, to a single worker. As complex tasks are difficult to accomplish; thus, the tasks are paid higher than micro-tasks to attract workers Hirth et al. (2013). Various examples include article writing, proofreading a large text, natural language processing (NLP), and solving biological problems.

In crowd-powered applications, complex tasks with a coarse granularity are usually decomposed into smaller units of works, performed by multiple workers, and then recomposed from the

Table 2
Different tasks, characteristics, and description.

	Micro-Tasks	Complex Tasks	Macro-Tasks	Creative Tasks
Properties	Simple, easy and require general skills, perform by individual workers	Difficult, often decomposable, involve workflow, require knowledge and skills of multiple workers	Non-decomposable, require expert skill and knowledge, often collaborative	Ideation, creativity, designing, co-creation
Examples	Image labelling, object recognition, audio transcribing, rating, ranking, and verification tasks	Nutrition estimation system, searching tool, data generation and labelling, writing recognition, text improvement	Defining research methodology, formulation of R&D approach Schmitz and Lykourentzou (2018), drafting documents, handwriting recognition Dai et al. (2013)	Software development, innovative projects, design competition of T-shirts, logos, 3D and graphic designs, solving medical mysteries
Platform/ workflow	MTurk, FigureEight, microTask, microWorkers	PlateMate Noronha et al. (2011), DataSift Parameswaran et al. (2014), Turkomatic Kulkarni et al. (2012), Turkit Little et al. (2009)	Upwork, CrowdSpring, OpenIDEO, TAS Schmitz and Lykourentzou (2018)	TopCoder, InnoCentive, 99designs, Threadless, CrowdMed, IdeaStorm
Participation	Individual (Collective)	Individual (Aggregative)	Collaboration (Iterative)	Competition (Selective)

answers of individual workers. In some instances, tasks are decomposed automatically using machines Little et al. (2010). Sometimes task decomposition is performed manually by the requesters Bernstein et al. (2010); Kulkarni et al. (2011). In this scenario, a requester splits the task into simple tasks and releases to the platform. After receiving results from workers, the requester recomposes the contributions to get the final answer. The tasks might also be decomposed by the workers Kittur et al. (2011). In this scenario, a task is submitted to the platform, the workers decompose the task into simple tasks, perform the tasks, and finally combine the results to get the outcome. The task composition might also be a combination of both manual and automatic strategies Kittur et al. (2012); Kulkarni et al. (2012). The workflow of task decomposition describes the control flow among sub-tasks, i.e., how the sub-tasks are chained together. Several workflows are proposed so far, such as sequential, parallel, iterative, recursive and hybrid. The selection of a suitable workflow for sub-tasks is of great importance as it can directly affect the quality of aggregated outcome Allahbakhsh et al. (2015).

3.3.3. Macro-Tasks

The decomposition of complex tasks to smaller sub-tasks may not always be feasible for all types of complex problems, as some tasks are non-decomposable complex tasks. The non-decomposition is a property of certain types of complex tasks, which cannot be handled through workflows used for decomposable complex tasks. Many real-life applications, such as survey feedback, writing a story, drafting documents, and many R&D problems, are non-decomposable tasks and cannot be divided into smaller sub-problems. These tasks differ from micro-tasks in that they require expert skills, may take more worker time, assume varying degrees of knowledge over a domain, and often involve multiple workers‘ collaboration, i.e., workers build on each other's contributions. This type of crowdsourcing is referred to as expert crowdsourcing Retelny et al. (2014) and the involved non-decomposable complex tasks are referred to as 'macro-tasks' Chittilappilly et al. (2016); Schmitz and Lykourentzou (2018). Some dedicated CS platforms for such tasks are *UpWork* Upwork (0000), *Crowdspring* Crowdspring (0000), and OpenIDEO OpenIDEO (0000).

Implementation of different types of workflows for decomposable and non-decomposable complex tasks is discussed in Section 6.

3.3.4. Creative task

The creative tasks are the earliest form of tasks in the form of competition and design contests that requires a diverse set of skills and knowledge. These tasks depend on individual creativity and uniqueness related to idea generation, creative design, or co-creation Chiu et al. (2014). Normally, requesters crowdsource creative tasks with limitations in finding the required solution to a particular problem. As far as incentives are concerned, the benefits of workers can be heterogeneous; however, the tasks are paid better for providing original and creative solutions. Expert validation and auction strategies can also be applied to select the winner for some creative tasks Tranquillini et al. (2015).

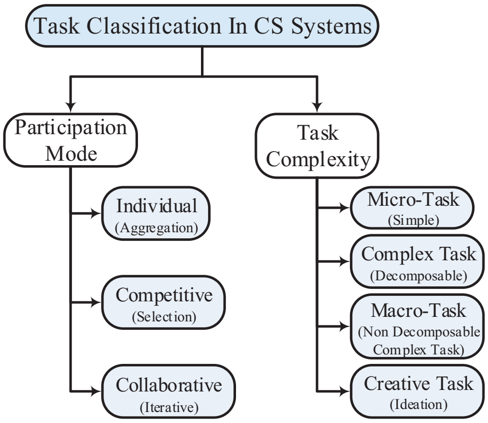

Task Classification In CS Systems
Participation Mode
Task Complexity
Individual (Aggregation)
Competitive (Selection)
Collaborative (Iterative)
Micro-Task (Simple)
Complex Task (Decomposable)
Macro-Task (Non Decomposable Complex Task)
Creative Task (Ideation)

Fig. 6. A taxonomy of task types in crowdsourcing systems. We characterize tasks along two main dimensions: participation mode and task complexity.

The crowdsourcing for creative tasks is a way of accessing the creativity of individuals. The contribution of workers can be both *selective* when the requester selects among the available solutions from proposed by workers, or *collaborative*, when the inputs from workers are pooled together (e.g., Existing notable CS platforms handling such tasks are *InnoCentive* InnoCentive (0000), solving R&D problems and innovative projects, *Threadless* Threadless (0000), holds design competition for T-shirts, *Quirky* Quirky (0000), for creative product design applications, *99designs* 99designs (0000), designing logos, 3D graphics, and *Wikipedia* Wikipedia (0000).

A taxonomy of task types in crowdsourcing is shown in Fig. 6 and a summary highlighting features of different task types are available in Table 2.

3.4. Task duration

The task duration specifies for how long the tasks remain active. After this time, the task is automatically stopped by the system. It is the acceptable time between uploading and submitting a task Saremi and Yang (2015). During the designing of a task, the requester needs to classify the time frame to complete a task, whether open-ended or limited. In most cases, the duration is finite, i.e., not open-ended, and the workers have to finish the task within the time duration specified by requesters. The time duration of tasks ranges from minutes up to a day or more Alt et al. (2010). For example, in 99designs 99designs (0000), some typical tasks stay open for four to seven days. However, requesters can specify the time frame as open-ended depending upon the task and workers' motivation.

3.5. User interface

The user interfaces are involved in implementing tasks in crowdsourcing systems. A CS system has two groups of users, i.e., requesters and workers, who interact with the CS system through user interfaces (UIs). Based on the user groups, there are two types of interfaces (UIs) named as *requester-UI* and *worker-UI*. Requesters design and deploy tasks using the requester-UI, whereas workers perform and submit answers of tasks using the worker-UI. A brief description of both interfaces is given below.

Requester-UI refers to the interface through which requesters access the platform and set up tasks. A requester can create and configure tasks, retrieving answers, and others using different types of interfaces including manual, template-based, specialized and proprietary interfaces.

(1) *Manual*: Requesters can build tasks manually. The manual designs ask the requesters to write HTML code using Command Line Tools (CLT) for web-based forms, commonly used for a small number of micro-tasks.

(2) *Template*: These are generally used for a large number of micro-tasks, and collection of results. By using UI templates, requesters design different types of tasks like image tagging, categorization, data collection tasks and others.

(3) *Specialized UI*: Requesters can design more complex tasks using specialized UIs. MTurk supports many UIs including task creation, blocking/unblocking of workers, collection of final submissions, and statistical analysis for both requesters and workers.

(4) *Proprietary UI*: Requesters build their servers to design and embed tasks in the existing platforms.

Most CS platforms provide both the manual and template-based interfaces for the instantiation of tasks. For example, MTurk Amazon Mechanical Turk (0000) provides predefined templates including classification, tagging and transcription. Many programmatic APIs exist in MTurk including the creation of HITs, collection of answers, blocking/unblocking workers. MTurk also provides a way for requesters to embed the tasks from her proprietary servers into MTurk using innerHTML. Other CS platform such as FigureEight FigureEight (0000) has similar functionalities with Mturk but differs in that it has a quality-control component (i.e., qualification test Zheng et al. (2017b)) to block low-quality workers. Upwork Upwork (0000) provides the richest APIs among CS platforms for updating contracts, managing payments, and extracting workers' statistics. Samasource samaSOURCE (0000) does not offer a UI in designing tasks but allows requesters in creating tasks programmatically or upload spreadsheets.

Worker-UI refers to the interface using which workers perform tasks for providing their contributions. The interface can be a *standard HTML form, web-UI, an API*, or *any specialized type* of UI system. User interface involves forms and textual descriptions in the interaction form, for example, typing in a text box, radio buttons, selecting in a drop-down menu, or taking a picture. Workers can login, search from the available lists of tasks, select tasks they are willing to attempt and execute them. The selection of a particular UI is highly non-trivial as the quality of interface design determines how effortlessly workers can perform tasks Marcus et al. (2012). User-friendly interfaces attract more workers and increase the possibility of quality contributions Allahbakhsh et al. (2013). The studies Finnerty et al. (2013); Rahmanian and Davis (2014) indicated that reducing cognitive load by eliminating unnecessary features from the interface designs help workers focus on the job and perform better in giving higher quality results. New interface design is proposed in proofreading digitized text in the crowdsourcing context Murano (2018). The design intended to improve performance, web usability, and decreasing the cognitive load of workers.

As discussed earlier, crowdsourcing involves different types of task. Thus, for each type of task, a requester needs to select an interface in such a way that workers view the relevant information and provide answers to tasks swiftly. For example, consider the task of entity resolution Vesdapunt et al. (2014) of determining if the photographs refer to the same person. The workers are provided with multiple photographs and required to group the photos for the same individual. While performing the task, can the worker label the photos with a group number? Or can she move them on the screen? Does the requester allow the worker to zoom into photos to check the details? Is the worker allowed to search other related photos on the web while performing a task? Does the worker have an option to indicate about her choices that she is certain or uncertain? The choice of these design interfaces impacts the performance as well as the quality of answers to tasks. All these choices even impact in recruiting and retaining workers, as tasks that are more interesting and enjoyable to attract workers.

Many existing works showed that user interfaces could affect the behaviour of workers Kittur et al. (2008); Yuen et al. (2011). A simple interface makes it easier for spammers to exploit the system, while complex interfaces with a confusing background and unstructured layout discourage genuine workers and could cause delays. Also, complex interfaces make it difficult for workers to accomplish tasks, or even annoying to workers in other cultures, and people might not join CS platforms Yin et al. (2014). Even the motivated workers may produce low-quality outputs or wrong data while interacting through a poorly designed interface Catallo and Martinenghi (2017). Therefore, interface designs for workers should be simple enough to avoid delay for efficient collection of inputs and highly robust against cheaters and spammers Allahbakhsh et al. (2013). For interacting workers, specialized user interfaces namely ‘negative suggest' interface is developed to avoid duplicate answers from workers by allowing them to view the garnered contents of other workers Trushkowsky et al. (2015). The study in á Campo et al. (2019) designed guidelines for intuitive and robust user interfaces to support designers in evaluating and comparing CS platforms.

4. Task settings

The requesters need to set up some parameters and settings during task preparation, depending on the general and particular requirement of the task. These parameters are task attributes, worker estimation and task interference.

4.1. Task attributes

Several parameters are required to explain a task. The requesters define these parameters at the creation time of the task. These parameters or attributes are used by the requester to provide an overview of CS activities. The workers can view these attributes before starting work and decide whether to perform the task according to the compensation, deadlines or others. It is of the view that requester should provide a longer description to perform complicated tasks. The system assigns a unique identification number (i.e., Task_ID) to each task by which it can be traced during ex-

Table 3
A list of attributes for a crowdsourcing task.

Domain/Element	Description
Task_ID	System generates unique ID
Task definition	Any specific group or name
Type / Category	Type or category of task
Keywords	Important words in the task
C_Date / C_Time	Creation date and time
E_Date / E_Time	Ending date and time
Location	Worker location (optional)
Limitations	Any constraints by worker(s)
Incentives	Rewards or other benefits
Task_DOC	Any document attached by crowdsourcer
Task_DES	Details of task / question from worker
NUM	# of worker(s) required to perform task
Req_Proof	Document or proof required from worker
Result_DOC	Any type of document attached by user
# of registration	Register users to participate in the task
# of Submissions	Numbers of solutions submitted
Duration	Duration between start and end date/time
# of Winners	Winner(s) status upon task completion

ecution within the platform. The other essential attributes are descriptions, task types, deadlines, rewards and likewise. A requester uses these attributes to define tasks explicitly for successful performance, whereas, the working ability to indicate how well a worker performs the task is also associated with these attributes Pu et al. (2017). Similarly, there are worker attributes that represent multiple aspects of the past performance in contributing to the current task. A requester can select workers based on these attributes. Previous studies Murturi et al. (2015); Yang et al. (2008) have provided a few attributes of the task; whereas, in this study, a consolidated and comprehensive list of task attributes is presented in Table 3.

4.2. Worker estimation

It is important to select the right number of workers before releasing a task Liu et al. (2012a). A different number of workers are required to perform various types of task. Depending on the size and complexity, the estimation of workers for a specific task is challenging. A requester can estimate workers for budget aware tasks. Worker estimation can be classified on the basis of number of worker(s) into *single worker task, multi-worker task* and *open call*. In single worker tasks, the worker is assumed as trusted without any mala fide intentions. Such tasks are difficult for computers and yet simple for humans. For example, labeling and reading CAPTCHAs are independent and straightforward tasks, require a single worker Allahbakhsh et al. (2015); Xintong et al. (2014). The study in Carvalho et al. (2016) empirically investigates the optimal number of workers for different types of tasks.

The success of crowdsourcing is based on the wisdom of the crowd, and a single worker may be unfair or biased for some specific tasks. So, it is more beneficial to design task for multiple workers. Writing an essay or proofreading a large text, multimedia data annotation are examples of such tasks which require multiple workers Good and Su (2013). Practically, all workers are not trusted, even so, it is rational to assume the majority of workers as trustworthy. Also, asking multiple workers to attempt the same task increase cost, since more workers need to be compensated. The answers from the workers are aggregated through different aggregation techniques, discussed in Section 9. In the open call scenario, everybody is allowed to participate in performing a task such as ESP game Weber and Robertson (2009) and Threadless Threadless (0000).

4.3. Interference

In marketplaces like MTurk, requesters often group a set of items into a batch of a single task. For example, a group of labeling tasks is performed by workers together in a batch. This has multiple advantages in saving time and cost, that is, workers read instructions once and perform a set of tasks all in one go, requesters are not required to provide instructions and pay for each task; requesters can attract more workers by paying higher amounts. Moreover, the online recruited workforce may include genuine workers, amateurs, cheaters and spammers (worker characteristics are discussed in Section 7). The performance of workers in a batch can be affected due to various characteristics of the crowd referred to as *interference* between different responses of tasks presented together to workers. The interference is identified in the form of groupthink Rosen (2011), cheating Eickhoff and de Vries (2013); Yuen et al. (2011), human bias Bonabeau (2009) and in-batch bias Zhuang and Young (2015).

The improper behavior of the crowd is important as task submissions can be compromised if a significant share of contributors is mainly interested in maximizing their monetary rewards by submitting generic and non-reflected answers rather actual answers Eickhoff and de Vries (2013). The authors in Eickhoff and de Vries (2013) demonstrated that implicit crowd filtering through task design is a superior means to control cheating. They concluded that cheaters are less frequently encountered in tasks that involve creativity and abstract thinking. In crowdsourcing, tasks can be negatively influenced by human biases. Various human biases such as self-service bias, availability bias, and self-confidence bias are discussed in Bonabeau (2009). The study recommends that collective-intelligence approaches can help mitigate the effects of human biases. Another issue is batch bias, as there can be more lenient or stingier workers in a batch to perform tasks. The study in Zhuang and Young (2015) identified bias on batches of multiple tasks simultaneously presented to workers and demonstrated that biasing might exist in binary classification and annotation tasks. To correct in-batch bias, various debiasing techniques are discussed in Zhuang et al. (2015). These interferences are important and need to take into account in designing and implementing CS systems.

5. Incentives

Incentives are essential for compensating people in performing tasks. Designing suitable incentives and compensation policies can affect the performance of workers and the output quality of the CS systems. These are a kind of incitement that encourages one to work hard. Without any incentive, it is unlikely that a person is interested in performing tasks. During the initialization phase, a requester defines different types of incentives. Numerous studies discussed incentives and different factors, but the pioneering work on worker motivation and incentives was carried out by Hackman and Oldham in 1976 Hackman and Oldham (1976). Different incentives in crowdsourcing include payment, task autonomy, skill variety, task identity, human capital advancement, and pastime Pilourdault et al. (2017). The incentives in CS systems are mainly classified as *intrinsic incentives* and *extrinsic incentives*. Intrinsic incentives refer to enjoyment and interest in the task itself. In contrast, extrinsic incentives are an instrument for achieving a certain desired outcome in the form of monetary or non-monetary benefits. A taxonomy of incentives in crowdsourcing systems is presented in Fig. 7.

5.1. Intrinsic incentives

The intrinsic incentives are associated with the process to control and maintain physical and mental activities. People are not always looking for monetary compensation. A person performs for

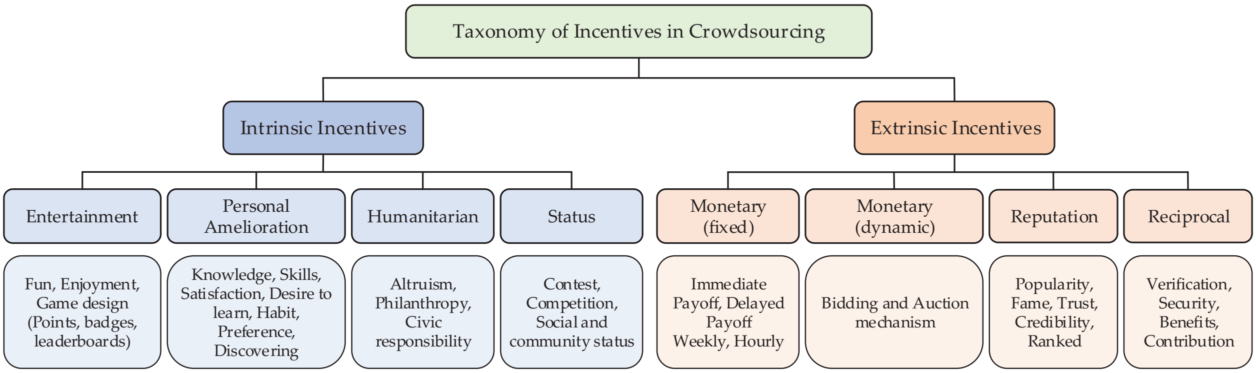

Taxonomy of Incentives in Crowdsourcing
Intrinsic Incentives
Extrinsic Incentives
Entertainment
Personal Amelioration
Humanitarian
Status
Monetary (fixed)
Monetary (dynamic)
Reputation
Reciprocal
Fun, Enjoyment, Game design (Points, badges, leaderboards)
Knowledge, Skills, Satisfaction, Desire to learn, Habit, Preference, Discovering
Altruism, Philanthropy, Civic responsibility
Contest, Competition, Social and community status
Immediate Payoff, Delayed Payoff Weekly, Hourly
Bidding and Auction mechanism
Popularity, Fame, Trust, Credibility, Ranked
Verification, Security, Benefits, Contribution

Fig. 7. A taxonomy of incentives in crowdsourcing.

fun, joy and involves challenges rather because of some pressure, prods or rewards. In Ryan and Deci (2000). Intrinsic incentives are defined as: *"Doing of an activity for its inherent satisfactions rather than for some separable consequence"*. The online community perform tasks to blow out thirst for intellectual curiosity. Common incentive dimensions are entertainment, personal development, learning and philanthropy. Below, various types of intrinsic incentives are briefed.

Natural. Entertainment and Learning: People wish to be entertained in doing interesting tasks, rather solving an instance of a complex problem. A requester can effectively increase user engagement and contribution by designing tasks in more entertaining ways. Gamification is a technique to engage workers in a non-gaming context by applying video game design principles Morschheuser et al. (2016). The gamification strategies help in improving users contribution in making tasks attractive, engaging and more fun by creating game-like interactions. Points, badges, leaderboards, stories, and avatars are common game design elements that are used in gamified CS systems Seaborn and Fels (2015); Hamari et al. (2014).

Gamification produces useful metadata as a by-product. By taking advantage of people desire to be entertained, problems can be solved efficiently by online game players. Games With A Purpose (GWAP) von Ahn and Dabbish (2008) is a popular and interesting way of solving complex tasks through games. The ESP game, a.k.a. Google image labeler Image Labeler - Crowdsource - Google (0000) in which players are randomly paired for labeling online images von Ahn and Dabbish (2004). A multi-player game-based approach *FoldIt* Cooper et al. (2011) in which players compete to optimize protein folding. The reCAPTCHA system von Ahn et al. (2008) acts as a CAPTCHA leveraging the crowd to transcribe words in digitizing old books and newspapers. While the platform *Dotsub* DotSUB (0000) collects video captions from crowd.

Personal Amelioration. Knowledge and Skill: Some people prefer to achieve satisfaction or enhance their creativity skills. When taking part in such tasks, people can improve their skills, feel satisfied and get feedback of self-worth. Also, learning and improving skills are important reasons for people to participate in such tasks. With *Duolingo* Duolingo (0000) people learn simultaneously new languages and contribute in translating web contents. One can improve photography skill through *istockphoto* iStockPhoto (0000) and at the same time contributes to develop stock photographs. With the *GalaxyZoo* Land et al. (2008), people study astronomy and help in discovering new galaxies through astronomical photographs. An image-to-text translation application *UbiAsk* Liu et al. (2012b) participants improve their language comprehension by tagging and translating images provided by foreign travelers. Other examples include collaborative knowledge aggregation like Wikipedia Wikipedia (0000) and Quora Quora (0000), which help to get awareness and knowledge by crowd community.

Humanitarian. Philanthropy and Altruism: During disaster and emergencies, people spontaneously offer help to trapped and victims. Workers perform tasks for social goodwill. For example, during the earthquake in Haiti Anderson et al. (2018) and Typhoon Haiyan, Philippines Chowdhury et al. (2016), volunteers and experts offered support and assistance in rescuing and relief efforts. Civic responsibility urges everyone for contributing data to make the society they live a better place. Another example, Audubon wildlife census in which workers perform in counting birds to conserve bird community (http://audubon.org/conservation/science/christmas-bird-count). Such intrinsic incentive could be ethical in providing satisfaction for contributing a noble cause involving humanitarian and religious goals AlShehry and Ferguson (2015).

Status. Competition and Identity: Competition is an essential factor that attracts people in performing tasks to get community identity. For instance, participants compete in designing T-shirts through an online competition by Threadless Jiang et al. (2015). Another instance is the collection of more than 100,000 photos through a contest by the PhotoCity Tuite et al. (2011), for generating 3D building models.

5.2. Extrinsic incentives

These incentives entice people to perform tasks. Such incentives include monetary reward, reputation, recognition evoked by financial and social reasons. Extrinsic incentives are defined as: *" to do work or refer to behaviour that pertains for something apart from and external to the work itself to attain some separable outcome such as monetary reward or recognition from other people"* Ryan and Deci (2000). The importance of payment compared to other rewards reveal that people are not always interested in performing tasks for fun and pastime Silberman et al. (2010). Extrinsic incentives positively moderate the relationship between task effort and engagement Liang et al. (2018). Below, various types of extrinsic incentives are briefed.

Monetary Incentives (Fixed): Monetary incentive is the best way to compensate workers, and most of CS platforms adopt this approach Teodoro et al. (2014); Singla and Krause (2013); Dubach et al. (2011). A requester pays to workers as 'Immediate Payoffs' upon the completion and acceptance of the task. For example, AMT Amazon Mechanical Turk (0000) and Squadhelp Squadhelp (0000) immediately pay to workers for successful completion of HITs. Some platforms pay for workers as 'Delayed Payoffs' that is used to generate future material benefits Kaufmann et al. (2011). Another way is that workers are paid through advance hiring and kept on hold Bernstein et al. (2011). Financial benefits are the most straight forward way to compensate people. These incentives significantly increase crowd participation and contribution along with better completion rate and processing time. However, at the same time, monetary rewards are significant as large payments may cause cheating and small payments are worse than nothing Gneezy and Rustichini (2000).

Monetary Incentives (Dynamic): Monetary rewards of micro-tasks generally include few cents or dollars, which are specified and fixed without any margin of negotiation Tranquillini et al. (2015). However, for the high-value complex tasks involved in enterprize CS systems, requesters are aware of the budget but cannot estimate the rewards. For example, a task like developing software projects, a crowdsourcer is unable to estimate the exact reward amount. Thus, rewards are tailored and customized by using *Auction* mechanisms, in which monetary incentives are dynamically adjusted through negotiation according to the real-time status. CS platform dynamically adjusts the price following the demand and supply relationship in the bidding process. The auction allows the crowdsourcer to publish task and ask workers to express reward for tasks; they are willing to perform Satzger et al. (2013). Auction is an efficient and cost-effective method to charge requesters and reward workers Qin et al. (2017). Incentives of CS platforms such as Freelancer, Guru, and oDesk are based on auction mechanisms with bidding and estimated budget Huang et al. (2017). Through the reverse auction, a market-based incentive mechanism is designed in Tian and Huang (2018). The model stimulates workers for performing tasks in unpopular areas and reduces the reserve price of tasks in popular areas.

Reputation. Fame & Trust: Active and devoted workers need recognition of their work to flourish reputation. Thus, reputation is a significant incentive to prompt user contributions as people perform tasks for social recognition and appreciation. Some could participate to get credibility or to get high rank among their peers in achieving mental satisfaction. It is also possible that people who stop receiving recognition and attention tend to stop contributing Wu et al. (2009).

Reciprocal Service: Tasks can be reciprocal with mutual benefits for crowdsourcer and contributors. Unlike many tasks, in some cases, workers receive a service exchange instead of financial compensation. For example, while downloading or opening an account at a website, a user is asked to complete a short survey or to solve a puzzle such as CAPTCHA. Being a reciprocal service, the puzzle is used for web security as graphical passwords determine whether or not the user is human. The service also contributes to solving a portion of a massive problem (e.g., digitization of out-of-print books by converting image handwriting into text) Von Ahn et al. (2003); Zhu et al. (2014), involves no money changes hand.

Others. Credits & prizes: There could be other incentives given to workers like T-shirts Tuite et al. (2011), snacks Heimerl et al. (2012), credit points Quora (0000); stackoverflow (0000), shopping vouchers or discount codes Aris and Din (2016) and get awards from a lottery draw Zheng et al. (2011) as well. Also, a bonus reward can be given to the best performer for increasing her overall reward.

6. Task decomposition

Most real-world problems are complicated and hard to solve that involve computational operations. These tasks require considerable efforts and dedicated resources, which limits the number of potential workers due to high skill barriers. Thus, decomposition techniques are used to split the main task into a chunk of subtasks, processed individually, and results of subtasks are recomposed to get the final solution. The decomposition techniques are commonly practised in computation field to reduce complexity. After decomposition, workers execute these subtasks dependently or independently to achieve parallelism. This atomization has several benefits in reducing the risk of failure, giving more flexibility of what and when work gets done to workers and requesters. A key aspect of task decomposition is that it enables parallelism and reduces the time required to perform the task, but may increase the cost and communication overheads. The decomposition of complex decomposable tasks is implemented by requester and platform or through crowdsourcing the task. Depending on the complexity, numerous techniques adopted for task decomposition such as *sequential, parallel, recursive, iterative*, and *hybrid* implementations. Below, we discuss different decomposition workflow techniques.

6.1. Sequential implementation

In this method, a task is divided into subtasks. The subtasks are performed serially, taking the output from the previous subtask as the input of later. The CS system selects the first subtask, executes it and then move towards the second subtask. This process continues till the last subtask is executed and then compiles result accordingly. The subtasks are interdependent as the output of one stage becomes the input of the second stage, and the next subtask could not start until the completion of the previous one. The inter-dependency is characterized by the feature that the output of a worker is the input of the following worker. The output quality depends on the efforts of workers who perform subtasks. This method is also referred to as ‘vertical task decomposition’ for dependent subtasks Jiang and Matsubara (2014). Various instances of sequential implementation used are given below.

SQA, is the first semantic parsing Sequential Question Answering (SQA) dataset Iyyer et al. (2016). Workers created the dataset by decomposing complicated questions into a sequence of multiple simple questions. It provides genetic information access and reduces the burden on users.

DynSP, a novel dynamic neural semantic parsing framework Iyyer et al. (2017), for answering complex questions. The model computes based on the state-action problem and learns modular neural network structures through a reward-guided search for training purposes.

State Machine, a workflow technology introduced in Zheng et al. (2017a), that automatically decomposes complex tasks and manage dependencies between subtasks. The system combines state machine workflow with the Price-Divide-Solve (PDS) algorithm. The PDS guides workers in decomposing complex tasks into micro-tasks. Using this model, many complex tasks are automatically decomposed with higher speed and quality.

PlateMate is a crowdsourcing based nutrition analysis system which estimates food intake and composition from photographs of food Noronha et al. (2011). The task of calculating calories in a food plate decomposed into a sequence of subtasks, i.e., *tagging, matching*, and *measuring* calories for each identified food and achieved reasonably accurate results.

6.2. Parallel implementation

In this technique, all subtasks are independent of each other and performed simultaneously. As the subtasks are executed in parallel, thus involve low time cost. The output quality depends only on the worker who executes it, contrary to sequential execution where performance depends on multiple workers who perform sequentially to produce output. The platform must synchronize and wait for the results of subtasks to collect the contributions for the task completion. The strategy is also referred to as ‘horizontal task decomposition’ for independent subtasks Jiang and Matsubara (2014). Various instances of parallel implementation used are given below.

DataSift is a crowd-powered search tool kit Parameswaran et al. (2014) that supports a large number of user queries. The approach is comprising of three steps, i.e., *discover, translate*, and *retrieve*. The workers perform the tasks independently and return results to the DataSift.

SCRIBE, a web-based interface Lasecki et al. (2012) that enables real-time captioning for the deaf and hard hearing people. Multiple workers perform individually and provide answers simultaneously, which are later combined by the system.

PhotoCity, an online game is developed Tuite et al. (2011), in which players collaboratively participate and independently capture multiple photos to construct 3D-models of buildings.

reCAPTCHA is a method von Ahn et al. (2008) in which users simultaneously perform to digitize the human knowledge available in old printed material through a word by word transcribing.

6.3. Recursive implementation

Complex tasks have dependencies, variable requirements and demand diversified expertise. Most real-world tasks are complex and cannot be decomposed easily like writing an essay, proofreading a large text, audio transcription, and data collection Allahbakhsh et al. (2015); Chen and Yu (2016). Workflows are required that facilitate decomposing of a complex task, managing subtasks interdependencies and aggregating the results. A price-divide-solve algorithm or divide-and-conquer algorithm, recursively decompose a complex task into small subtasks. The decomposition process continues till further splitting is not possible, or the desired granularity is achieved. Every subtask is executed by recursively applying similar decomposing, followed by aggregating their results to obtain the outcome. The recursive process generates multiple decompositions across several steps and makes task performance efficient and accurate Chittilappilly et al. (2016). Various instances of recursive implementation used are given below.

Turkomatic is a CS tool designed in Kulkarni et al. (2012) that solves complex tasks. The system harnesses workers to perform a continuous PDS loop to recursively divide a complex task into subtasks until an atomic level is achieved to solve subtask easily. The other workers are asked to verify the results and combine at a later stage.

CLTE Corpora is a Cross-Lingual Textual Entailment corpora created using crowdsourcing Chen et al. (2015). The system used the divide-and-conquer approach to fragment complex task involving the generation and annotating entailment pairs into simple subtasks, efficiently performed by workers.

In Trushkowsky et al. (2015), a statistical estimation technique is used to divide set space into disjoint partitions that effectively increases speed and answer diversity in CS enumeration queries. In Zhang et al. (2011), by using divide-and-conquer the workers serve as data oracles, decompose problems that enable task solution via human computation. Recursive task atomization is used in Amato et al. (2013), to recognize handwriting that not only increased the accuracy but also obtained the solution in a faster way. In Breazeal et al. (2013), the divide-and-conquer approach is used in performing collaborative tasks.

6.4. Iterative implementation

In this method, a complex task is decomposed into multiple subtasks where the input of later subtask is taken from the output of the previous subtask. The workers collaborate and perform using iterative workflow. Decision theory and Artificial Intelligence (AI) are applied to construct and optimize the iterative workflow. The number of iterations is dependent on the voting mechanism or the budget beforehand such that workflow stops upon exhaustion or a stopping condition is met. An iterative workflow is a powerful tool as workers perform collaboratively and make improvement based on each other work. The disadvantage of the iterative process is their long completion time. Various instances of iterative implementation used are given below.

TurKit is a tool kit for deploying iterative tasks that use AMT workers Little et al. (2009). The complex task of handwriting recognition is implemented through an iterative workflow in which succession of workers performs HITs that build on each other's work. The starting point of the new iteration is chosen through transcription receiving the majority vote. To meet with the challenge of long-running iterations, a *crash-and-rerun* programming model is introduced in TurKit Little et al. (2010). The new model met the system challenges and also reduced the time cost.

CrowdCode is a procedure for software development LaToza et al. (2014), that decomposes programming task into dynamically generated micro-tasks. The model iteratively splits a high-level task into several low-level tasks and uses artifacts to coordinate the work. State machine function is used for iteratively generating micro-tasks.

6.5. Hybrid implementation

A hybrid workflow is tailored by combining different decomposition techniques for complex and highly interrelated tasks such as writing a blog or an article. Therefore applying a mixed workflow can improve efficiency. If the output of two parallel tasks is the input of the following task, the system must synchronize and wait for both parallel tasks to complete before advancing to the next task. Several typical hybrid workflows are presented as follows.

Soylent is a word processing interface Bernstein et al. (2010) that enables writers to shorten, proofread, and edit documents. The system uses a crowd programming approach *Find-Fix-Verify* (FFV) that splits the open-ended task into a sequence of three subtasks. Workers identified text in the first phase and assigned to other workers for fixing in the second phase, and in the final phase, the results are verified. The main tasks are performed sequentially, whereas the shorten subtask is performed with multiple iterations.

CrowdForge, is a general-purpose approach based on micro-task platforms Kittur et al. (2011). The framework accomplishes complex interdependent tasks and provides typical functions of partitioning, mapping, and reduction as the basic building blocks for distributed process flow. A complex task is broken up systematically and dynamically into sequential and parallelizable subtasks and merged back later. The limitation of workflow for complex tasks is that the decomposition structure is static and fixed by crowdsourcer.

Control Group Approach is a crowdsourced validation process based on sequential and parallel decomposition techniques Hirth et al. (2013). The system performs segmentation of the task into multiple subtasks. The received subtasks in the first step are assigned to workers for evaluation, and in the last step, answers are generated. The system performs subtasks in parallel during the first phase, which reduces both time and cost consumption.

6.6. Macro-Task

Some real-life applications are macro-task, having a coarse-granularity level, referred to as *non-decomposable* tasks. These are context-heavy data processing tasks and require expert skills, more time, involve inter-task dependency and needed specific domain knowledge Garcia-Molina et al. (2016). Few instances of macro-tasks are entity resolution, survey feedback, transcribing a speech, and document editing Schmitz and Lykourentzou (2016); Li et al. (2017a). Macro-tasks are hard to split because the task

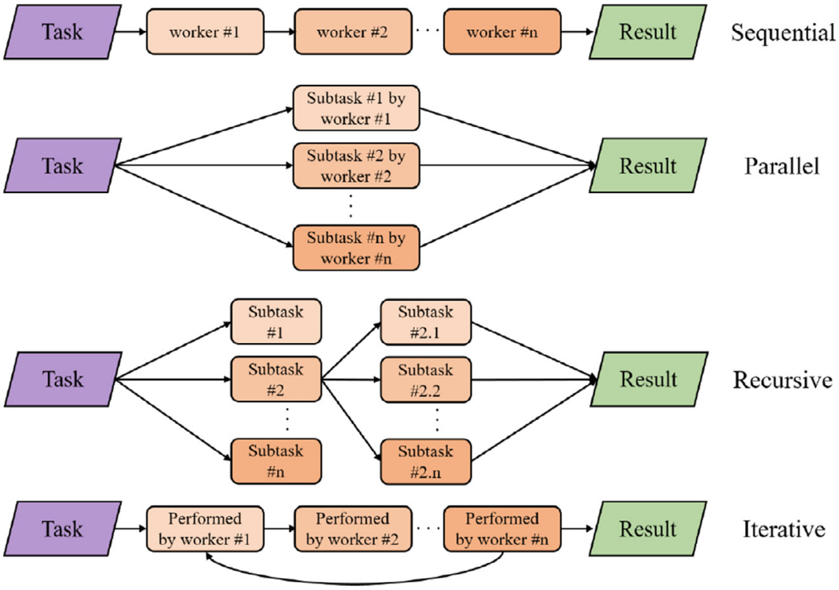


Fig. 8. Task decomposition techniques.

loses its context information upon splitting, therefore accomplished through multiple workers and involve high incentives. Generally, workers are not willing to perform the macro-task as it may take a long time to complete Li et al. (2017a). Some renowned CS platforms deal with macro-tasks are *UpWork* Upwork (0000), and *Crowdspring* Crowdspring (0000). Available workflows for macro-task are as follows.

Argonaut is a framework that combines domain-independent automated models to accomplish macro-task Haas et al. (2015). The system organizes workers hierarchically, enabling trusted workers to improve the output of less-experienced workers. It provides predictive error model and effectively identifies tasks that need corrections. The model tracks the workers' answers to put them at the top of the hierarchy. As a result, the system uses existing automated systems having no requirement of decomposed tasks.

TAS is an expert crowdsourcing model of Task Assignment and Sequencing (TAS) Schmitz and Lykourentzou (2018). The system sequentially improves non-decomposable macro-tasks through workers sequencing across time slots until a sufficient quality level is achieved. The model is based on worker sequencing across time slots. A given macro-task is assigned to an expert worker for a specific time slot. The quality of the task is assessed against a predefined level. The task is assigned to the next worker if the threshold is not achieved. Multiple macro-tasks can be performed simultaneously with multiple workers through different time slots.

A task is split into subtasks, performed either in sequential, parallel, recursive, iterative or hybrid and later combine the results into a coherent solution Ikeda et al. (2016). Beyond these efforts, people also customize workflow of task decomposition for their specific applications and needs. A vision on potential decomposition techniques in CS systems is presented in Fig. 8 and key points are available in Table 4.

Table 4
Characteristics of Decomposition Implementations.

Implementation	Key points
Sequential	• Ease to work out • Stepwise process visibility • Substantial output after each step • Large completion time • Workers dependency delays • Require careful determination of steps • Error propagation and amplification
Parallel	• Fast execution • More budget required • Synchronization required • Difficult implementation
Recursive	• Fast execution • Recursive delay • Complex implementation • Efficient algorithm implementation
Iterative	• Easy implementation • Large execution time • Careful determination of no. of iterations • Subsequent iterations can improve result • Workers dependency delay
Hybrid	• Efficient execution • Customized strategy • More execution time • More budget required

7. Finding crowd and platform selection

Searching human workers is a significant issue in CS systems; workers are available in the existing marketplaces or requesters can recruit workers defining their component of user management. Whereas, for the execution of tasks, many CS platforms with different services and features exist. Based on task requirements and priorities, requesters can select a platform in performing tasks. A brief description of finding crowd and platform selection is given below.

7.1. Finding crowd

Human workers for accomplishing tasks are available at the existing virtual labor market (VLM) like MTurk or UpWork. In some individual cases, a private crowd may be preferred. In such a private crowd, workers are recruited by the company (or a third party) that is interested in working on CS tasks. Requesters can devise their recruiting component in the CS system. For instance, a recruiting function is defined as 'user management' that contains registration, evaluation, group formation, and coordination Hetmank (2013). The crowd finding in crowdsourcing is required to consider the dimensions of *task requirement* and *worker characteristics* Ross et al. (2010).

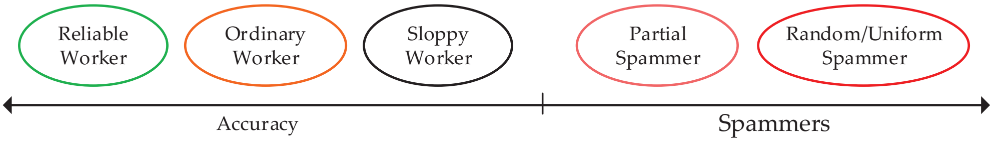


Fig. 9. Visualization of worker types in crowdsourcing.

Task requirements: One of the key challenges is to find workers that are appropriate for our tasks, particularly if the task require particular skills or involve complexity. Importance of worker recruitment and negative impact on quality when workers randomly recruited are discussed Kazai (2010). For example, in Threadless Threadless (0000), people require prior knowledge about the technologies (e.g., graphics editing software) in performing design related tasks. Similarly, workers require basic knowledge of camera parameters in performing tasks of visual questions Bigham et al. (2010). For the tasks with specific requirements, a requester needs to find workers having specialized knowledge and skills as ordinary workers may be unable to perform those tasks.

Depending on task complexity, different types of tasks may require different amounts of workers to perform. Sometimes, only a single worker is required in performing some tasks; in other cases, a team is necessary to perform other tasks. For *micro-tasks*, a requester needs to find individual workers who can perform the tasks independently with little computational efforts. The *complex tasks* are first fragmented into sub-tasks using different decomposition techniques. Then, requesters need to find individuals who can perform decomposed sub-tasks within a short time and atomic computation effort. Requesters are required to recruit groups in performing the *macro-tasks*, as individuals may be unable to perform such non-decomposable tasks. To perform the *creative tasks*, requesters can recruit either individuals or teams, as individuals and others may perform some tasks are required a team of workers. Thus, the requirement of workers for different types of tasks in crowdsourcing can be classified as *individual-oriented vs team-oriented*.

Team formation is an activity of finding a group of workers having complementary skills for performing tasks in a cooperative way Huang et al. (2017). The team formation mechanism, namely TruTeam Liu et al. (2018a), is designed for collaborative crowdsourcing market. Different team-based factors, such as worker-worker affinity and comfort level, are considered for collaborative crowdsourcing Rahman et al. (2015). A team formation problem was studied in Lappas et al. (2009), in which a team of expert workers is formed to perform tasks according to the skills and relationships of users in social networks. According to the control mechanism, team formation is categorized into *self-organized* and *centralized* approaches Rokicki et al. (2015); Liu et al. (2018a). In self-organized strategy, individual worker self-form a team using local visibility in the crowd, whereas, in a centralized approach, the team formation is made based on skills or bid price from a team to complete the designated task. The CS system or the requester has full rights in controlling the team formation in a centralized approach.

Worker characteristics: The other important aspect in finding crowd is worker characteristics. Crowdsourcing is an online activity in which workers are recruited without any interaction or interview. Moreover, people belong to different demographics or personality traits in terms of location, gender, age, income, qualifications, skills and others. Thus, users in crowdsourcing can be categorized into six types, including reliable workers, ordinary workers, sloppy workers, uniform spammers, random spammers, and partial spammers Liu et al. (2018b) and depicted in Fig. 9. Among the crowd, the presence of cheaters, spammers, and incompetent workers with proper workers makes it difficult to identify whether or not a user is a cheater or a legitimate worker. These spammers and cheaters having ulterior motives can sabotage a task or attempt to give fake responses for immediate financial gains. Spammers are opportunistic who often submit random answers without understanding a task or sometimes create automated ‘bots’ to pretend as a human worker. The malicious users work individually or in groups focusing on tasks to provide unsatisfactory answers, spread rumours to affect the market Boutsis and Kalogeraki (2014).

Finding a proper crowd for performing the right task by the right worker is key to the success of the CS system. Workers may be recruited based on their initial evaluation before participating in the CS process and reputation after they have completed some tasks. The former applies to the workers' profile in which skill levels and expertise are determined through pre-qualification tasks, gold questions or entry questions at the time of recruitment Corney et al. (2010). Whereas, the latter based on the validation of completed tasks Mashhadi and Capra (2011). The worker reputation is discussed in Section 11. To judge workers' characteristics, the workplace such as UpWork allows interviewing potential workers, whereas others need certain special identification techniques Upwork (0000).

7.2. Platform selection

Crowdsourcing platform is an online venue or a website that provides an interface between requesters and workers to manage the life cycle of tasks. A platform provides services to the requester to submit task and workers to perform a task. There are many CS platforms, whereas *Amazon Mechanical Turk*, (AMT) Amazon Mechanical Turk (0000) is one of the first recognized and popular platform among them. Based on applications, CS platforms are categorized into *general purpose* and *specialized* platforms Chittilappily et al. (2016). Further, these platforms are classified into *stand-alone* and *meta* platforms Daniel et al. (2018). The former applies to the platforms which have developed their processes and do not depend on other platforms to perform tasks, whereas, the latter depends on previously mentioned stand-alone platforms and act as a proxy for other platforms. AMT is a general-purpose, stand-alone CS platform that can be used for micro-tasks and FigureEight FigureEight (0000) is a general-purpose meta-platform act as proxy toward other platforms. Many such CS platforms developed like CloudFlower, oDesk, CrowdGuru, 99designs, Topcoder, Innocentive, CrowdForge, UpWork, Zhubajie and many more, have differences of technical features and operational characteristics but the overall crowdsourcing process is the same for these platforms. Several studies exist Kucherbaev et al. (2016); Tranquillini et al. (2015); Chittilappily et al. (2016); Daniel et al. (2018), discussed different CS platforms. Here, we only give a summary of key features for some popular platforms as available in Table 5.

8. Task assignment

Task assignment is an important aspect in crowdsourcing, as poor assignment affect the system performance in terms of time and money. The main problems in task assignment are unavailability of workers' information and continuous arrival of both tasks

Table 5
A summary of key features of some CS platforms.

CS-platform	Key features
AMT	• Popular and firstly recognized in CS market • Specialized in paid micro-tasks • General purpose stand-alone platform • Highest number of globally sourced workers • Provision of worker qualification tests • Allows complete task management to requesters • Reward tactics implemented • Available information about HIT ◇ Requester, description, keywords, qualification required, Reward, expiration date, allotted time, available HITs
FigureEight	• Suitable for paid simple jobs and processes • General purpose meta-platform • Acts as proxy toward AMT or Samasource • Mainly used by enterprizes and data scientists • Sophisticated interfaces to create tasks • Workers evaluated by gold units • Reward tactics implemented • Allows partial task management to requester
99designs	• Mainly contest based design and ideation tasks • Specialized stand-alone platform • Monetary rewards for the best solution
CrowdForge	• Mainly for complex (decomposable) tasks • General purpose meta-platform • Services rely on AMT
UpWork	• Mainly for complex (non-decomposable) tasks • General purpose stand-alone platform • Auction based for freelancers

and workers at the platform. Some platforms do not publish workers information, but systems require this information for security reasons during worker recruitment. Based on information accessibility, task assignment in crowdsourcing is classified into *Online* and *Offline* assignment methods. Task assignment strategies in which workers information is not known upfront, and each task needs to be assigned immediately are online assignments, whereas methods in which workers information is available as prior are offline assignments.

8.1. Online task assignment

In the online assignment, workers actively visit the platform, and no prior information of workers is available with the crowdsourcer. A requester provides task information like budget, the total number of questions, and other information. The tasks are assigned to the available online workers upon arrival at the platform. The challenge is that requester does not have any information about the skills of unknown workers. Highly dynamic and transient behavior of workers is another problem. Initially, skills are not available but are learned later through observations upon completion of tasks. This unavailability of data for processing may rise to the problem of data uncertainty. The assignment is also referred to as a worker-centric approach.

Online assignment supports immediate response; however, the optimal solution is not feasible due to the dynamism of tasks and workers. Various online task assignment strategies are discussed such as AskIt Boim et al. (2012), Quality-aware Task Assignment system for Crowdsourcing Applications (QASCA) Zheng et al. (2015), Online Task Assignment (OnTac) Yang et al. (2016b), Online Joint Inference and Assignment (OJIA) Carusi et al. (2018), and LEarning Automata-based Task assignment (LEATask) Moayedikia et al. (2018) are as follows.

AskIt: A real-time system for interactive crowd data-sourcing applications that efficiently performs question routing to appropriate workers and reduces the uncertainty of collected data. The algorithm is developed to solve the optimization problem considering two uncertainty metrics and four different constraints for user and question. The entropy-like method is used to measure uncertainty, whereas collaborative filtering is used to predict the missing answers. With the help of a collaborative filter, the system provided high-quality predictions and also reduced sparsity for large datasets. This system not only minimizes the number of questions but also reduces the uncertainty in the data distribution.

QASCA: It is an application-driven online task assignment framework, comprised of two evaluation metrics, *Accuracy* and *F-score*. These matrices significantly improved the quality of results through distribution matrices. Two linear-time algorithms are used to address ground truth problem, which reduced cost, complexity and maximize the answer quality. The workflow prompts two activities from workers, HIT request and HIT completion. The attempted questions are excluded, and new questions are updated for onward assignment. Based on these constructions, the evaluation metric *F*-score is maximized, and the question set is assigned to workers. The answers and quality parameters are populated in the confusion matrix, which updates the current distribution matrix. Task allocation is carried by gradually assigning tasks to appropriate workers with a high probability of correct answers. The system is developed on top of AMT, evaluated for five CS applications and found to improve quality and accuracy within a fixed budget.

OnTac: The algorithm is based on a probabilistic model and makes an online estimation of labeler expertise, question difficulties and ground truth. The model estimates parameters using the Expectation-Maximization (EM) algorithm, which consists of two phases, i.e., the assignment phase (E-step) and the inference phase (M-step). Assignment phase chooses a suitable question based on estimated parameters and asks the label from current labeler. In inference, online estimation is made for question difficulty, labeler competency, and ground truth. General EM framework adopts an iterative approach, where each iteration takes E-step and M-step. In the *E-step*, it computes the distribution of posterior on given labels derived from previous M-step. In the *M-step*, it computes system parameters by maximizing log-likelihood function based on the current estimate and received labels from E-step. The convergence is achieved iteratively; thus system grows slowly than linear growth and achieves high accuracy with minimum budget.

OJIA: The online framework finds the correct answer of categorical tasks like binary questions or annotations. The system adaptively assigns questions to appropriate workers in a round-based scheme by jointly learning workers' abilities and questions' difficulties. This adaptive assignment maximizes the chances of correct answers. The system used a statistical model to integrate question difficulty and worker ability. For adaptive assignment and aggregation of responses, the system used the inference technique and probabilistic graphical model. The framework is generalized in predicting correct answers of multiple-choice questions using expectation propagation algorithm.

LEATask: A new task assignment algorithm that works based on workers' performance similarities. The model consists two phases, i.e., *exploration* and *exploitation*. In the exploration phase, some training tasks are assigned to workers and learned about their reliability using Groot reliability estimation algorithm. Then, the workers clustered into groups using hierarchical clustering algorithm based on reliability similarities. After that, the algorithm associated each cluster with a learning automata (LA) and trained each automaton using data of its cluster. The exploitation phase used the trained automata and the clusters to assign tasks to the newly arrived worker.

Different features of online task assignment techniques are summarized in Table 6.

Table 6
Characteristics of online task assignment techniques.

Techniques	Key points
AskIt	• Efficient question routing • Minimizes sparsity and uncertainty • Highly interactive • Not optimal for unanswered questions
QASCA	• Addresses ground truth problems • Focusses application types • Accurate measures of worker quality • High computation cost
OnTac	• Focusses on ground truth estimation • Achieves high accuracy & efficiency • Cost effective, optimizes labeling budget • High processing time
OJIA	• Focusses accuracy prediction of correct answer • Adaptively perform assignments • Jointly consider worker ability & question difficulty • Execution with unknown number of rounds • Challenges of scalability with efficiency
LEATask	• Unsupervised task assignment • No reliance on ground truth • No reliance on workers' dynamic characteristics • Robustness to sparseness • Deals only with binary classification tasks

8.2. *Offline task assignment*

In the offline assignment, a CS system has complete knowledge of workers and tasks. A requester submits tasks and users send their profiles to the platform before starting the activity. The requester has maximum control over assignment as knowledge, skill, and a bid of worker are all known. Accordingly, the system publishes a task by selecting workers to maximize profit. The offline assignment is performed based on global knowledge of all tasks and workers; thus, it is optimal in minimizing cost assignment. However, this strategy is not feasible in real-time scenarios, as significant delays are involved. This assignment is also referred to as a platform-centric approach. Various offline task assignment strategies are discussed such as Iterative Algorithm (Iter. Algo.) Karger et al. (2011), Offline Approximation Algorithm (OAA) Assadi et al. (2015), Collaborative Approach (Collab. Appr.) Rahman et al. (2015), Task Assignment with Quality Requirement (TAQR) Yin et al. (2017) are as follows.

Iter. Algo.: The iterative algorithm is designed to assign tasks with absolute reliability under budget constraint. The model constructs a bipartite graph with task and worker as nodes, and the edges are associated with the task assignment. A requester specifies the number of assigned tasks to each worker as the right degree and workers for each task as the left degree. A random graph generation method *configuration model* is used to generate a bipartite graph. After each iteration, the messages are updated, and results are improved through inference algorithm that uses task messages and worker messages. Reliability of worker is represented by the message of that worker. The final estimate of the correct solution is calculated upon submission of all answers. The designed algorithm achieved an optimal solution with a high computation cost.

OAA: The algorithm assigned different tasks to workers for which workers bid, arrival order and cost of each task are known upfront. The system considered adversarial setting and random permutation to model worker arriving sequence. Workers and their corresponding tasks are represented using a bipartite graph. When a worker begins a task, an edge is drawn connecting two nodes. The source node is connected with all workers, and the target node is connected with all tasks. The edges between the nodes correspond to task allocation, and flow represents the budget for assigning tasks. For computing output, the system has subroutine Fixed Threshold Policy (FTP), that assigns a predefined value to function. The algorithm executes for all workers and assigns a task to the idle worker having bid below the predefined threshold. Finally, it arbitrarily stops execution when any unassigned task is found, for which worker bid is equal or less than a predefined value.

Table 7
Characteristics of offline task assignment techniques.

Techniques	Key points
Iter. Algo.	• Minimizes budget • Efficient execution • Achieved answer reliability • Provides the optimal solution • High computational cost
OAA	• Easy to implement • Focus on homogeneous workers • Provides optimal solution • Assumption of bid to budget ratio not consistent
Collab. Appr.	• Task assignment in collaborative environment • Optimized to maximize quality & minimize cost • Maintain group integrity after decomposition • Approximation with provable guarantees • Skills shortage within subgroups
TAQR	• Focus on both requester and worker requirements • Constraints: workers heterogeneity, task quality • Formulated as many-to-one matching problem • Improved task success ratio and worker happiness • NP-hard problem with acceptable time complexity

Collab. Appr.: Task assignment optimization is investigated for collaborative crowdsourcing by incorporating team-based factors. The model considers both individual and group-based human factors like worker skill, wage and team-based factors of *worker-worker affinity* and *upper critical mass*. The former describes the comfort level of a worker in performing the same task with peers, while the latter describes the maximum allowed group size of workers. Groups of workers are formed with complementary skills and expertise in solving complex collaborative tasks. Another factor of *intra-group affinity* estimates workers collaboration effectiveness within the group. When group size increases beyond upper critical mass, the group is divided into small subgroups. The subgroups are assigned a portion from the primary task and contributions are integrated from each subgroup to recompose the main task using *inter-group affinity*. An efficient approximation algorithm is proposed with provable guarantees. A collaborative sentence translation was successfully performed using AMT.

TAQR: Task assignment is performed with guaranteed quality for CS platforms. The assignment strategy is formulated as a many-to-one matching problem, in which multiple workers are assigned to a task resolving conflicting interests of workers and requesters. The workers are heterogeneous in their skill levels, and tasks have different quality requirements with budget limitations, are all known to the crowdsourcer. The proposed stable matching model is based on the Deferred Acceptance (DA) algorithm, that significantly improved the task success ratio and worker satisfaction with adequate time complexity.

Different features of offline task assignment techniques are summarized in Table 7.

9. Answer aggregation

As mentioned before, crowdsourcing leverages the intelligence and wisdom by outsourcing tasks to potentially large groups of people for contributions. Generally, a task is assigned to several workers to provide redundancy and collect the wisdom of the crowd. The basic idea is to assign a task to multiple workers and infer its answers by integrating contributions from all workers. Such assignment leads not only to multiple solutions for the same task but also reduces the impact of wrong answers and worker biasing. Despite the availability of redundancy, it is required to merge and process contributions to have accurate, rational and unique result Luz et al. (2015). This critical step is related to answer veracity and final performance of the overall CS process.

Answer aggregation can be either manual or automatic. A requester can manually aggregate the workers' answers for simple and easy tasks. Commonly used aggregation methods are *filtering, wisdom of the crowd (WoC)*, and *voting mechanisms*. Filtering methods are prominent in CS and aim to filter out bad items such as poor responses and unqualified workers. For instance, Marcus et al. (2012) uses gold standard questions (whose true answers are known) to filter outputs, whereas Abraham et al. (2016) filters workers based on their reputation. The WoC Surowiecki (2005) is an alternative to filtering and considers responses from all workers; therefore, need to compensate more workers. In the voting mechanism, the answer with the highest number of votes is finalized as the aggregated answer. FigureEight FigureEight (0000) implemented both filtering and voting schemes to select correct results and filter workers with wrong answers.

The aggregation depends on worker selection mechanism to filter and assess workers' expertise, giving prominence to workers with high expertise. AMT Amazon Mechanical Turk (0000) implemented filtering mechanism to evaluate workers' skills and expertise. Another way is to crowdsource the aggregation process as an answer aggregation task. Systems like CrowdForge Kittur et al. (2011) and Jabberwocky Ahmad et al. (2011) make use of crowdsourcing for answer aggregation. A taxonomy of answer aggregation for micro-task CS platform is proposed in Moayedikia et al. (2017). Based on computational model, answer aggregation techniques are broadly classified into *non-iterative aggregation* and *iterative aggregation* techniques. We discuss and analyze different state-of-the-art answer aggregation techniques as follows.

9.1. Non-Iterative aggregation

Non-iterative aggregation techniques use heuristics to compute the individual result of each task. Commonly used techniques include the application of voting and filtering process Luz et al. (2015). Various non-iterative aggregation techniques are discussed such as Averaging Output (Avg. O/P) Surowiecki (2005), Weighted Majority Voting (WMV) Littlestone and Warmuth (1994), Majority Voting (MV) Kuncheva et al. (2003), HoneyPot (HP) Lee et al. (2010), Expert Label Injected Crowd Estimation (ELICE) Khan Khattak and Salleb-Aouissi (2011), and Bayesian Voting (BV) Li et al. (2016) are as follows.

Avg. O/P is a simple technique to compute a single aggregated answer that averages the outputs of multiple workers working on the same task. Adding multi-worker redundancy and averaging outputs can be close to the actual answer, but the cost increases in compensating more workers. In the book "The Wisdom of Crowds" Surowiecki (2005), the author has argued that averaging information in a large group, resulting in decisions, are often better than could have been made by any individual of the group. The WoC examines several instances of crowd wisdom such as ox weight estimating experiment, gaming sports betting spread, and Columbia shuttle disaster Brabham (2008), has shown that averaging the answers of multiple people can produce overall accurate aggregated answer.

WMV is used in CS systems for improving accuracy as the scheme considers each worker quality. The mechanism is also known as *supermajority* Castano et al. (2016), which assigns a weight to each worker according to its trustworthiness. The higher voting weight is assigned to the higher quality worker and answer with the highest aggregated weight is declared as the final task result. WMV is used in Liu and Liu (2017); Chowdhury et al. (2015); Li et al. (2017b) for various CS applications.

MV, is a simple solution based on majority voting for aggregating workers answers and computing trusted labels for truth discovery. It is a commonly used method which selects the highest votes for the final aggregated value of a question without any preprocessing. However, in practice, the system is ineffective and have failure chances as it does not account for knowledge, skill, standard errors of workers as well as task difficulty Liu and Liu (2017). The limitations of MV are that it assumes all workers as trusted and predictably unsuccessful when the majority of untrusted users vote for same task Zheng et al. (2017b).

HP, performs identically to MV except for the preprocess filtration, used to eliminate unreliable users. The filtering prevents hackers, cheaters, untrustworthy and unskilled workers to enter into the CS process. Filtering is carried out by placing a set of questions with known answers in other questions. The users who fail to give correct answers of specified questions are declared as spam candidates and removed accordingly. Possible labels for remaining workers are computed using MV model Kuncheva et al. (2003). The disadvantages are the unavailability of filtering questions and misidentification of some truthful workers as cheaters if the questions are too difficult Albarqouni et al. (2016); Shafiee et al. (2016).

ELICE, is an extension of HP and used for accurate data labeling. It filters questions to determine two latent attributes of *user ability* and *question difficulty*, and then combine these attributes in the aggregation process. The worker expertise is calculated by the ratio of correct answers to filtering questions and toughness of each question is estimated by the workers who provide correct answers of filtering questions. The logistic regression model computes the probability of the correct answer. Briefly, ELICE achieves high veracity for result aggregation as the method considers both user ability and question difficulty. All responses are weighted with expertise and difficulty; the probabilistic answer to be correct is adjusted well. However, it has the same drawback as in HP, where a trusted user might be misidentified as spammer Abassi and Boukhris (2016); Poonkuzhali and Mohana (2016).

BV, is another voting scheme which uses Bayes' theorem and considers worker quality to compute the probability distribution of each response. For the results having randomness, the Randomized Bayesian Voting returned an answer with some probability and improved the error bound for worst-case analysis Li et al. (2016).

9.2. Iterative aggregation

Iterative aggregation techniques perform a series of iterations and yield high-quality results in combining tasks. These techniques consist of two stages. First, the aggregated value of each question is updated regularly. Second, worker expertise is adjusted based on the worker's answer Quoc Viet Hung et al. (2013). These methods encompass a set of incremental computation where the probability of correct answer is incremented after each round. The techniques are based on incremental rounding mechanism, so question filtering is not required. Despite the advantage of high accuracy, a significant problem is high running time as many iterations are involved in convergence Quoc Viet Hung et al. (2013); Abassi and Boukhris (2016). Various iterative aggregation techniques are discussed such as Expectation-Maximization (EM) Dawid and Skene (1979), Supervised Learning from Multiple Experts (SLME) Raykar et al. (2009), Generative model of Labels, Abilities, and Difficulties (GLAD) Whitehill et al. (2009) and Iterative Learning (ITER) Karger et al. (2011).

EM, is a probabilistic model that requires no ground truth Li et al. (2017a). It is used to infer predicted aggregated answers and estimate worker reliability. This technique computes the probabilities of labels in two phases: *E-phase* and *M-phase*. In E-phase, label probabilities are estimated given worker reliable accuracy degree.

Table 8
Characteristics of Aggregation Techniques.

Technique	Trapping Set	Aggregation Model	Worker Expertise	Question Difficulty	Model	Probability
Avg.	no	non-iter	no	no	online	no
WMV	no	non-iter	no	no	online	no
MV	no	non-iter	no	no	online	no
HP	yes	non-iter	no	no	online	no
ELICE	yes	non-iter	yes	yes	offline	yes
BV	no	non-iter	yes	no	offline	yes
EM	no	iter	yes	no	offline	yes
SLME	no	iter	yes	no	offline	yes
GLAD	no	iter	yes	yes	offline	yes
ITER	no	iter	yes	yes	offline	no

In M-phase, worker expertise is estimated again since the current probability of each label is available from E-phase. This iterative process continues until all label probabilities are unchanged. Briefly, EM formed a confusion matrix for computing aggregated results by exploiting worker accuracy. Long computation time is a critical problem, as many iterations are required to aggregate labels Abassi and Boukhris (2017). EM is efficient and outperforms than MV in the presence of hidden variables. In Yang et al. (2016b), EM is used for adaptive online task assignment. The EM-based framework is proposed in Artikis et al. (2014), that facilitates heterogeneous streaming in urban traffic management. Improved EM-based approach is employed Zhang et al. (2018) for answer aggregation considering workers characteristics and task difficulty.

SLME, is another probabilistic approach Raykar et al. (2009). It performs like EM, but instead of using error or confusion matrix, it characterizes the expertise of workers through two well-known statistical measures of *sensitivity* and *specificity*. These are ratios of correctly assigned positive and negative answers, respectively. Both statistical measures are defined only for binary labels; thus, a major drawback is its incompatibility with multiple labeling.

GLAD, is the synergy of ELICE and EM. It formulates the probabilistic labeling process, which considers worker ability and question difficulty. It leverages the inference method to infer expertise of each worker, difficulty of each question and the most probable label for each task Georgescu and Zhu (2014). The model captures two cases; first, high expertise worker has a higher probability of giving the correct answer when multiple workers respond to a question. Second, the probability of getting the correct answer of the toughest question is low when a worker responds to multiple questions Quoc Viet Hung et al. (2013).

ITER, is inspired by the Belief Propagation (BP) algorithm. Like other models, it estimates worker expertise, the difficulty level of question and yield a high-quality result in combining tasks. Unlike other systems for which reliability of all answers by a worker has one value (which is expertise level), it estimates the reliability of each answer independently. Also, each question difficulty is estimated separately. The worker expertise is computed as the sum of question difficulty weighted with associated answer reliability. In Bernstein et al. (2011), the crowd, is recruited to find the best final photo in a short video and aggregated the result after several iterations. ITER is a simple model as compared to EM; moreover, it is robust against random and malicious annotators. Being iterative, it involves heavy computations and lengthy processing delays.

For a new task, it is not very easy for users to select an appropriate technique. A benchmark tool *Benchmark for Aggregate Techniques in Crowdsourcing* (BATC) developed in Nguyen et al. (2013) that evaluate aggregation techniques using five performance metrics of computation time, accuracy, sensitivity, compatibility and error estimation. Researchers and software developers can use BATC to select and configure the best-suited aggregation technique for their potential task. Worth mentioning that WMV, MV, HP, ELICE, and BV have fast processing and preferred for low response time, whereas others are not suitable for large inputs. Moreover, iterative techniques (EM, SLME, GLAD, ITER) must repeat the whole process; therefore, recommended for offline analysis for which answer set is not changed.

Different features of aggregation techniques are available in Table 8.

10. Validation and rewards

Once the crowdsourcing process is over, results are reported to the requester by the CS platform. Then, the conclusion phase starts, in which a requester needs to validate the answers and compensate workers. The workers receive incentives according to rewarding strategies. Here we discuss different validation methods and rewarding strategies, whereas different types of incentives are already discussed in Section 5. Moreover, based on the outcome of the validation phase, a requester or platform can perform the worker reputation management, discussed in Section 11.

10.1. Validation methods

The objective of validation is to assess given inputs by the crowd. Depending on worker characteristics and nature of the task, the results of crowd submission are required to validate before rewarding. The answers of the tasks depend on the performance of sizeable anonymous crowd. Among the crowd, a user can be a reliable worker with in-depth knowledge to give reliable answers or a spammer who intentionally give wrong or carelessly give random inputs. Also, some tasks are assigned to redundant users, so multiple answers are available for the same task. This important activity of validation in the crowdsourcing process depends on the task type and available budget. Manual validation requires additional resources of time, budget and human activities. For reducing time and cost, crowd-based validation processes are designed in Hirth et al. (2013), that provides significance level for detecting cheating of untrustworthy users. Commonly used approaches are given below.

Design parameters: A requester can validate workers' answers using various defined fields and attributes in the task UI. These are standard guidelines for designing tasks with formal requirements such as avoiding the empty fields. It bounds the worker to select at least one option from the given, or workers only allowed to put data in the allowed formats, or by mentioning various mandatory and optional fields in the user page of task FigureEight (0000).

Decision Approach: To validate the answer, a requester crowdsource a validation task through the CS platform and ask workers to rank the answer. The output depends on the given rank of workers, which is acceptable if it has a value higher than a predefined threshold otherwise rejected by the CS platform. To validate the answer, two crowd-based schemes, namely Majority Decision (MD) and Control Group (CG), are introduced in Hirth et al. (2011). In MD, the overall result is valid even if some workers cheat the system. Whereas, in CG, the answer is validated upon meeting the threshold criteria.

Expert Validation: Expert validation is used in crowd answers to provide trustworthy answers and mitigate uncertainty in crowd answers. The validation can be performed by the requester or by some external expert. The expert is asked to assess the given answer and produce an assessment value. If the value is positive, the work is accepted otherwise rejected. Mostly, expert validation used for creative tasks such as a logo or color scheme design. Many CS platforms such as AMT include validation engine in the validation process to verify results Hung et al. (2015).

Matching: In this process, the validation requires matching of feedback of at least two players on a given answer. If two players immediately agree, the agreed-on value (yes or no) is chosen as the output of the process. Otherwise, a third player is asked to give her input which is treated as the output. The output surely matches with the input of one of the previous players.

Gold Questions: The validation of answers depends on gold questions with known ground truth. The answers of gold questions are used to estimate the validity without analysis of the given answers. The correct answers of the gold questions are compared with the threshold settings which decides the acceptance or rejection of full answers. The drawback is that a requester needs to pay workers for gold questions.

10.2. Reward policy

The incentive provisioning is a significant activity to motivate or discourage a worker before performing tasks. Once the tasks are performed, the validation stage serves to assess the performance of workers and provides input to the requester. Based on this input, a requester decides whether to reward workers for the rendered services or retribution for wrong-doings. Different reward strategies and tactics exist, here we discuss most using reward policies.

Pay to All / None: It is a trivial logic, in which payment is made to all participants or none of them, irrespective of the fact that a requester is satisfied or not from the task result. The CS platform automatically executes this option after workers perform tasks and results submitted to the platform. The approach is commonly applied to simple tasks such as labeling and other similar atomic tasks.

Upon Confirmation: It is the most commonly used strategy in which workers are rewarded after confirmation from a requester. A requester confirms based on the outcome of the validation stage. The scheme is popular for situations where the workers are rewarded only if a requester is satisfied with their work. The confirmation strategy is used mostly for rewarding, complex tasks.

The Best Solution: This scheme is generally adopted for tasks in which workers contest for the reward. Workers perform the same task and submit results to the requester where the top k-solution(s) (depending on crowdsourcer) are rewarded. The logic works well for creative tasks.

By the end of a successful crowdsourcing process, both the workers and the requesters are benefited from different ways and are discussed below:

10.3. Worker / requester benefits

Several benefits are identified for the workers and the requesters. Different types and forms of workers' benefits are already discussed in Section 5. Requesters can get pecuniary and non-pecuniary benefits in the form of a solution or an improved product, community good and many more. The other benefits can refer to the knowledge acquired, innovation, and creativity as compared with the traditional processes. The tasks are completed cost-effectively, with higher productivity, and through a diverse workforce available 24/7. A requester can also benefit from the relationship of growing networks of other stakeholders that are involved with the requester organization.

11. Reputation

In the crowdsourcing process, the trust relationship between the requesters and workers reflects the probability that the requester expects to receive a quality contribution from the workers and more workers to take part in for honest requesters. Below, the worker reputation and requester reputation are briefly explained.

11.1. Worker reputation

The reputation of a worker is an indicator of recommendation for participation in future CS tasks. It is related to the expertise in completing tasks that reflect worker's trustworthiness and can affect requesters hiring decisions. Based on the outcome of the validation process, a requester can accept or reject the task result, and this impacts the worker reputation. A worker can improve her reputation by contributing additional material and pieces of evidence during task execution. If a task is wrongly performed as determined in the validation phase, the worker receives a lower reputation, and ultimately fewer tasks and little to no money Silberman et al. (2010). Similarly, a requester can impose a social sanction, i.e., banning or blacklisting the worker for submitting lousy work.

The reputation score of workers is computed using *content-based* and *community-based* strategies Allahbakhsh et al. (2012b). In the content-based approach, worker reputation is evaluated based on the content analysis of the outcome of the task performed by the worker. For instance, the reputation of Wikipedia Adler and de Alfaro (2007) contributors determined by the quality of shared content. In the community-based approach, reputation based on the feedback received implicitly or explicitly from community members. Many online services like TripAdvisor, Stack overflow and Reddit, support community-based approach to evaluate the user reputation. Compared to the content-based approach, the community-based approach is not popular in CS systems.

A fine-grained reputation can be computed by considering other metrics and components such as bias in workers interest, allotted timelines, expertise, evaluator feedback and credibility, task completion rate and task approval rate. In AMT, the reputation of workers based on HIT approval rate and other qualification such as language skill. Subsequently, a requester can refuse worker whose task completion rate or task approval rate is below a predefined threshold. A reputation management model in Allahbakhsh et al. (2012a) adequately considers the credibility of evaluators, credit amount of tasks, and time of evaluation. Reputation score and expertise are the primary metrics of worker profile that need to be considered while recruiting workers. To mitigate the skewness and sparsity in the reputation score, a reputation mechanism based on Bayesian update is *WorkerRank* Daltayanni et al. (2015) that considers requesters implicit feedback rather explicit feedback after task completion. The worker reputation model in Jun and Shaoping (2017) based on the active degree of workers for recent and last month active days.

11.2. Requester reputation

The reputation of a requester depends on whether the task is well defined, whether the task is straightforward to perform, whether the monetary reward is fair enough given the task requirements, how promptly workers are paid, and how well the requester communicated with workers Sheehan (2017). There is no systematic reputation mechanism for requesters using which workers evaluate individual tasks. MTurk has not provided a rating system for workers to rate its requesters. Moreover, there is unbalanced power dynamics and asymmetry in the reputations of requesters and workers. As a result, several different online communities have been developed using which workers rate the requesters, tasks, and give their remarks.

One most successful community tool, the Tukopticon Irani and Silberman (2013), is a browser extension for Chrome and Firefox that augments workers' view of their task lists in MTurk with comments other workers have provided about requesters. Workers enter ratings of requesters they have worked with, their tasks, that include quantitative ratings as well as comments explaining their ratings. Turkopticon manages requester reputation using four attributes: *communicativity* describes the responsiveness to worker communications and concerns; *generosity* captures the kindness of a requester in paying for the amount of time to perform a task; *fairness*, tells the impartiality of a requester in accepting or rejecting answers (also studied in Allahbakhsh et al. (2012a)); and *promptness*, tells the swiftness of a requester for approving and paying the successful work. The other instance is Turker Nation Turker Nation (0000), an online discussion forum where workers can comment on requesters and about tasks. These external sites have a substantial effect on the acceptance rate of tasks and therefore serve effectively to report and avoid shady requesters.

There are a few instances where requesters could directly interact with workers. The MTurk interface allows workers to contact with requesters Mason and Suri (2012). For example, workers may wish to query task with the requester if part of their task is confusing or unclear. Similarly, workers can comment on online forums regarding different aspects of a task.

12. Challenges and future directions

As mentioned before, substantial research work has been performed in the crowdsourcing, but there still exist problems. In this section, we discuss the challenges related to different segments of the proposed framework of the crowdsourcing process together with task management, worker management, unexplored issues and future opportunities.

12.1. Task management

The significant issues in crowdsourcing framework are related to task management, i.e., how to design a task, how to break a task, how to assign a task, how to evaluate contributions, how to estimate reputation, and how to aggregate results to get the final solution.

12.1.1. Task design

Specific patterns and design choice can significantly increase worker efficiency by reducing errors. Depending on the type of task, user-interface, the design of a task can affect the performance of CS systems. Designing general purpose HITs for a wide variety of tasks with the specific requirement is still challenging, such as designers should consider intuitiveness, anchoring and pop-out dynamics Kulkarni et al. (2012); Rahmanian and Davis (2014). Existing works related to different aspects of task designing is available in Section 3. Different task design requirements make it challenging, and further research is required to distinguish task design strategies and their impact on performance metrics. Task design is a significant challenge for the crowdsourcers to get proper results from workers Chen et al. (2011). Significant efforts are still required to develop guidelines for the design of robust and intuitive task UIs for automated assessment methods, attempted in Miniukovich and De Angeli (2015).

12.1.2. Task decomposition and aggregation

Complexity. In crowdsourcing, complexity is identified as an important task property for the relationship between human and computer artifacts. It is one of the challenges in task decomposition to identify the right granularity (i.e., complexity level) of tasks. Practically, all tasks are neither equally easy nor difficult, rather closely related to worker ability. In fact, what is easy for one may be hard for others. Task implementation must limit cognitive complexity. Few empirical studies have demonstrated how task difficulty level can affect error rates. As the difficulty level increases, the estimated errors provided by workers also increases. Unfortunately, task complexity and worker ability are typically unknown upfront for designers. It has inferred that a task is difficult if workers end up with disagreement on the answer, and the worker has the ability if she gets a majority of answers as correct. To quantify and to understand the complexity of the task, a little effort is available in Yang et al. (2016a). This way of inferring needs to assume workers independence. As, if for workers correlated errors, auto-detection and auto-correction are a lot harder. Of course, the provision of theoretical performance guarantee is harder when complexity is considered for decomposition and aggregation.

Batching. Typically, requesters (like AMT) form a batch of questions about various items in a single task. Batching has many advantages, like a requester can attract more workers through high payments, workers can provide answers to many questions in one go and have more context awareness to task diversity. The major drawback of batching is bias within skewed data sets. There has been little work, except for Zhuang et al. (2015), in which the impact of and remedy for batch bias in crowd algorithms is studied so far.

With the rise of task-specific biases and errors, aggregating users' contributions are becoming a challenge. Both manual and automatic aggregation methods have developed. For instance, users manually combine edits for Wikipedia Wikipedia (0000), while ESP Weber and Robertson (2009) does so automatically. However, automatic aggregation still needs to be fully explored. Also, if aggregation algorithms do not work correctly, the requester may face financial loss. Another problem in aggregation is to decide what to do if users differ in their opinion for decomposed tasks. Further, many real-life applications are based on macro-tasks, which are hard to split and involve multiple workers. Upon splitting, these tasks lose the context information. Thus, splitting and aggregation are somewhat challenging.

12.1.3. Evaluation and reputation

Reputation management has significant issues from both users and evaluators. First, the primary determinant factor of workers' reputation is the outcome of legitimate workers inputs. Whereas CS systems have no information about workers, so it is hard to identify malicious users and rule out their inputs. The spammers and cheaters always try to and find some way in seeping through the system. Second, requesters do not give feedback to workers which gives an advantage to malicious users to continue working. Third, the rank-boosting scam, in which a worker creates a requester account, distributes and completes simple tasks from her worker login and boosts her rank in the system, is another challenge for CS systems. Finally, evaluators try to manipulate the reputation through random evaluations, dynamic behavior and biased attitude for specific workers.

Manipulation from dishonest evaluators can lead to harm worker community possibly causing unemployment and inadequate worker selection. Therefore, the system must be capable enough to deter and detect, block or remove these dishonest users. So, far little work done in the existing CS systems is already discussed; therefore, it is desired to address these issues adequately. Assessment methods are mostly applied after task execution, while they could also be applied before and during task execution to enable preemptive interventions. Suitable analytical features and interventions could improve the accuracy of outputs, cost efficiency and timeliness.

12.1.4. Macro-Tasks (non-decomposable)

Existing research mainly focuses on micro-tasks that can be easily handled for assignment and instantly performed by workers. Whereas, many real-life applications are based on macro-tasks such as transcribing speech, writing a paper and others. Macro-tasks are hard to split because the task loses its context information upon splitting. Generally, workers are not willing to perform the whole macro-task as it engages workers for a long time. Therefore, handling macro-tasks is somewhat hard and technologically challenging, including the task design, allocation to workers or machines, and aggregating the results. We indeed believe that the studies of macro-tasks and their workflows are an important research topic in the future.

12.2. Worker management

Identity and Ability: The real identity of users is one of the significant challenges in CS systems. It is not very easy to determine the intention of users; people participate in CS activities with different motives, whereas some people perform tasks only for monetary gains. Therefore, the financial incentives in CS lead to Internet fraud Almendra and Schwabe (2009). People using CS websites, do not care for moral and security principles, impose irreparable damages to the system such as Sybil attacks by injecting false data in the system Cox (2011). The framework proposed in Shafiee et al. (2016) maintains security and supervision in CS systems. Ethical culture is required, which includes fair pricing strategies and build a long-term career for workers Kocsis and de Vreede (2016). Generally, all workers are mostly alike in their abilities, working speed and error rates. However, practically, this is not true; different workers have different abilities and processing speeds. Considering these factors help to route tasks to appropriate workers. Ethical issues related to economics and knowledge exchange are required to explore for CS implications Standing and Standing (2018).

Attitude and Biasing: Mostly, CS systems do not consider human factors like attitude and biases Garcia-Molina et al. (2016). The irresponsible attitude gives a negative impact on CS. For instance, workers avoid doing work if they satisfice by doing less work. They give answers to known questions without caring for instructions or answer by random guessing. Workers are affected by fatigue and laziness when they have to answer similar questions. Human factor causes the challenge of skill declination, possibly due to boredom, annoyance and inactiveness Gadiraju et al. (2015). Biasing is another critical factor affecting CS systems. No significant research is available taking worker bias into account instead of ignoring worker specific bias by assuming a uniform error rate for all workers.

Recruitment and Retaining: Crowd recruitment always remains a challenge as workers with specific knowledge and skills are required for different types of task. Thus, recruiting the right users, encouraging and retaining them is an essential fundamental challenge Doan et al. (2011). Also, the recruitment of new workers negatively influences system latency. Moreover, the behavior of workers recruited online cannot be guaranteed. These are the potential threats for CS systems. Even though the right workers are recruited, the other big issue is how to attract them in performing tasks.

After recruitment, one should consider ‘how to retain workers’, as attracting people is easier than retaining them Difallah et al. (2014). Various encouragement and retaining policies exist that attempt to retain users such as providing instant gratification, enjoyable experience or service, a way to establish fame, ownership situations or set up competitions Doan et al. (2011). For any CS system, motivating and retaining users always remains challenging.

Industrial Rights: CS is considered as a biggest paradigm shift in innovation since industrial revolution Felstiner (2011). The tasks are often performed by anonymous workers having no physical working site. CS industry may face potential legal issues as legislatures and legal protection for crowdsourced employees are not available, and workers may be denied for the protection of employment laws Alqahtani et al. (2017). Crowd laborers are paid low payments for their cognitive work, enjoy no job security and earn no benefits. Indeed, venders may restrain them from doing so. Workers have no legal protection, and the cyberspace where they perform remains unregulated for labor laws Peng (2011). Additionally, they face complications with deception, privacy and information asymmetry.

In AMT and other lopsided platforms, workers have limited information on tasks and their employer. On the other hand, in UpWork Upwork (0000), firms and workers can interact or negotiate before task execution. Workers can propose their hourly rates and employer gain access to a qualified and stable workforce. Despite numerous advantages of CS, workers, particularly in paid crowdsourcing, are denied the protection of employment laws without much recourse to vindicate their rights Felstiner (2011).

Demographic Composition: Worker composition in VLM fluctuates with the time of day, day of the week, month and even year. Different accuracies and delays have observed for tasks when deployed at different timings. Anecdotal evidence suggested that these effects are periodic (i.e., delays on weekends are much higher as compared with weekdays), while delays and accuracies change over the year in different VLMs. These parameters have a strong influence on accuracy, cost and delay, and are challenging to validate the performance of crowdsourced solutions Garcia-Molina et al. (2016).

12.3. Unexplored issues

Several other challenges to crowdsourcing are coordination and communication, privacy, intellectual property (IP), security and copyright problem are still unexplored.

Coordination and Communication: Direct communication between crowdsourcers and workers can be engineered to share intermediary feedbacks and query task details. Coordination is a primary challenge in distributed systems oriented towards agents. As CS involves anonymous users, it becomes difficult to communicate and coordinate them in achieving a deep understanding of problems. Workers may perform tasks without understanding the problem. How well the task design is crafted, the challenge is one-sided communication through which little information is provided to the participants. Thus, workers are left to make assumptions which lead to off-track and out of scope solutions. All these add-up to waste time and money. Establishing coordination and its impact on performance is an important direction for future research.

Privacy: Various types of privacy problems are associated with crowdsourcing. Firstly, crowdsourcers require privacy of their tasks as it may contain some sensitive attributes. Anonymous users are joining the platform, and there exists a possibility that malicious users could reveal important information to the competitors, which is a considerable threat to requester privacy. Although crowdsourcer publishes anonymous data to workers such as K-Anonymity Wu et al. (2014), it may lessen the accuracy as workers cannot get precise information about tasks. Therefore, the trade-off between privacy and accuracy is challenging Durward et al. (2016).

Secondly, the database records (such as location, profession, and hobby) of workers is available at CS platform, or personal information of workers can be inferred from their answers, whereas workers ask for privacy-preserving. Also, requesters require personal information of the workers available for assigning tasks to knowledgeable and skilled workers. In this way, a requester put user privacy at risk Wolfson and Lease (2011). Unfortunately, developing effective strategies for protecting user privacy remains an open research problem. Thus, security researchers and practitioners alike must strive to develop better tools and techniques to help maintain privacy.

Intellectual Property ownership and security: In open call CS systems, everyone can access information. The requesters may have difficulties to describe a task entirely for security reasons. Limited information is provided to workers, whereas tasks should be described explicitly for early processing. Intellectual property issues occur during transferring of task deliverables Bauer et al. (2016). Users reveal innovative and potentially valuable intellectual property that could be a risk for imitation and theft. For example, a worker provided a highly relevant answer for some task. The requester decided to use the solution and later discovered IP violation. The violation of IP rights and misuse of personal information could face severe and disastrous consequences for social harm Peng (2011). Further, the rejected submissions of task remain under the possession of requester, lead to potential copyright ownership problems. Finally, infringement and cheating of IP and copyright are serious problems that adversely affect the development of crowdsourcing To and Lai (2015). For different CS applications, especially creative production and ideation approaches involve an online community submitting original ideas. Thus, it is required to have a policy for handling IP and copyright violations Brabham (2013b). A crowdfunding platform 'Kickstarter' was subject to a lawsuit involved in the 3D-printer project, which another company, '3D-Systems', claimed infringed on its intellectual property (http://bbc.com/news/technology-20434031).

The ultimate goal of crowdsourcing is yet to be achieved, but it is vital to make sure that crowdsourcing algorithms should decompose the tasks correctly, assigning the right task to the right worker, rule out spammers, consider intellectual property, security, copyright, and compile error-free answers. All the problems above are open research issues and need to explore in future. This paper presents the technical features of the framework of the crowdsourcing process and highlights gaps and opportunities for the researchers.

13. Conclusion

Crowdsourcing is an evolving phenomenon, recognized as an effective and efficient mechanism in solving distributed human-powered problems. The main advantage of crowdsourcing is cost reduction as crowdsourcer does not need in setting up the infrastructure and workers are always available to perform tasks. However, crowdsourcing still has many challenges due to its openness and unreliability. As the entire process is online, the crowdsourcer is unaware of whether a user is genuine or wicked. Like that, other challenges also exist in crowdsourcing.

In this paper, we have performed a contemporary survey of crowdsourcing techniques which are used to solve different problems. We first proposed a firm definition of crowdsourcing associated with crowdsourcing model. Then, we analyzed the current literature and discussed the weaknesses and strengths of the current work. Next, we introduced a general research framework of crowdsourcing which serves as a road map in understanding complete crowdsourcing process and activities to build traditional and futuristic systems. Subsequently, we discussed task designing to increase crowd contributions, properly structured incentives to motivate workers, and task decomposition and aggregating answers to increase the success rate. Different answer validation methods, reward tactics, and worker reputation strategies are illustrated that are widely used for many crowdsourcing platforms. The techniques and methods proposed by researchers are briefly described to address the challenges. We digress the details of interface designs, software, and formulae proofs to keep the work focused. Finally, we explored major open issues and identified possible research directions to stimulate future efforts for real-life applications.

The multifaceted coverage of this survey will open up new avenues of research in crowdsourcing. The outcome will help developers and researchers as a reference model to concretely and precisely state their particular interpretation, design implication, and configuration. We believe that our survey will provide guidelines for future research works in crowdsourcing.

Declaration of Competing Interest

The authors declare that they have no known competing financialinterestsor personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Shahzad Sarwar Bhatti: Conceptualization, Methodology, Writing - original draft. **Xiaofeng Gao:** Writing - review & editing. **Guihai Chen:** Supervision.

Acknowledgments

This work was supported by the National Key R&D Program of China [2018YFB1004703]; the National Natural Science Foundation of China [61872238, 61672353]; the Shanghai Science and Technology Fund [17510740200]; the Huawei Innovation Research Program [HO2018085286]; the State Key Laboratory of Air Traffic Management System and Technology [SKLATM20180X]; the Tencent Social Ads Rhino-Bird Focused Research Program, and the CCF-Huawei Database System Innovation Research Plan [CCF-Huawei DBIR2019002A].

References

áCampo, S., Khan, V.J., Papangelis, K., Markopoulos, P., 2019. Community heuristics for user interface evaluation of crowdsourcing platforms. Future Gener. Comp. Syst. (FGCS) 95, 775–789. doi:10.1016/j.future.2018.02.028.

Abassi, L., Boukhris, I., 2016. Crowd Label Aggregation Under a Belief Function Framework. In: Proc. Int. Conf. on Knowl. Sci. Eng. and Manag. (KSEM), pp. 185–196. doi:10.1007/978-3-319-47650-6.

Abassi, L., Boukhris, I., 2017. Iterative Aggregation of Crowdsourced Tasks Within the Belief Function Theory. In: Proc. European Conf. on Symb. and Quanti. Appr. to Reason. and Uncertain. (ECSQARU), pp. 159–168. doi:10.1007/3-540-44652-4.

Abraham, I., Alonso, O., Kandylas, V., Patel, R., Shelford, S., Slivkins, A., 2016. How Many Workers to Ask? Adaptive Exploration for Collecting High Quality Labels. In: Proc. Int. ACM conf. Res. and Dev. Inf. Retriev. (SIGIR), pp. 473–482.

Adler, B.T., de Alfaro, L., 2007. A Content-Driven Reputation System for the Wikipedia. In: ACM WWW, pp. 261–270.

Afuah, A., Tucci, C.L., 2012. Crowdsourcing as a solution to distant search. Academy of Manag. Rev. 37 (3), 355–375. doi:10.5465/amr.2010.0146.

Ahmad, S., Battle, A., Malkani, Z., Kamvar, S.D., 2011. The Jabberwocky Programming Environment for Structured Social Computing. In: Proc. ACM Symp. on User Inter. Softw. and Technol. (UIST), pp. 53–64. doi:10.1145/2047196.2047203.

von Ahn, L., Dabbish, L., 2004. Labeling images with a computer game. In: Proc. ACM Int. Conf. Human Factors in Comput. Syst. (CHI), pp. 319–326. doi:10.1145/985692.985733.

von Ahn, L., Dabbish, L., 2008. Designing games with a purpose. Commun. ACM 51 (8), 58–67. doi:10.1145/1378704.1378719.

von Ahn, L., Maurer, B., McMillen, C., Abraham, D., Blum, M., 2008. Recaptcha: human-Based character recognition via web security measures. Science (AAAS) 321 (5895), 1465–1468. doi:10.1126/science.1160379.

Albarqouni, S., Baur, C., Achilles, F., Belagiannis, V., Demirci, S., Navab, N., 2016. Aggnet: deep learning from crowds for mitosis detection in breast cancer histology images. IEEE Trans. Med. Imaging 35 (5), 1313–1321. doi:10.1109/TMI.2016.2528120.

Alkharashi, A., Renauld, K., 2018. Privacy in Crowdsourcing: A Systematic Review. In: Proc. Int. Conf. Information Security (ISC), pp. 387–400. doi:10.1201/9780203507223.

Allahbakhsh, M., Arbabi, S., Shirazi, M., Motahari-Nezhad, H.R., 2015. A Task Decomposition Framework for Surveying the Crowd Contextual Insights. In: Proc. IEEE Int. Conf. Service-Oriented Comput. and App. (SOCA), pp. 155–162. doi:10.1109/SOCA.2015.32.

Allahbakhsh, M., Benatallah, B., Ignjatovic, A., 2013. Quality control in crowdsourcing systems. IEEE Internet Comput. 17 (2), 76–81. doi:10.1109/MIC.2013.20.

Allahbakhsh, M., Ignjatovic, A., Benatallah, B., Beheshti, S.-M.-R., Bertino, E., Foo, N., 2012. Reputation Management in Crowdsourcing Systems. In: Proc. IEEE Int. Conf. Collab. Comput.: Netw. Appl. and Workshar. (CollaborateCom), pp. 664–671. doi:10.4108/icst.collaboratecom.2012.250499.

Allahbakhsh, M., Ignjatovic, A., Benatallah, B., Beheshti, S.-M.-R., Foo, N., Bertino, E., 2012. An analytic approach to people evaluation in crowdsourcing systems. CoRR abs/1211.3, 1–21.

Almendra, V., Schwabe, D., 2009. Fraud Detection by Human Agents: A Pilot Study. In: Proc. Int. Conf. on E-Comm. and Web Technol. (EC-Web), pp. 300–311. doi:10.1007/978-3-642-03964-5_28.

Alqahtani, B.A., El-shoubaki, R.T., Noorwali, F.A., Allouh, D., Hemalatha, M., 2017. Legal and ethical issues of crowdsourcing. Int. J. of Computer Appl. (IJCA) 167 (10), 1–3.

AlShehry, M.A., Ferguson, B.W., 2015. A Taxonomy of Crowdsourcing Campaigns. In: Proc. ACM Int. Conf. on World Wide Web (WWW), pp. 475–479. doi:10.1145/2740908.2741747.

Alt, F., Shirazi, A.S., Schmidt, A., Kramer, U., Nawaz, Z., 2010. Location-based crowdsourcing: Extending Crowdsourcing to the Real World. In: Proc. ACM Nordic Conf. Human-Computer Interaction (NordiCHI), pp. 13–22. doi:10.1145/1868914.1868921.

Amato, A., Sappa, A.D., Fornés, A., Lumbreras, F., Lladós, J., 2013. Divide and conquer: Atomizing and Parallelizing a Task in a Mobile Crowdsourcing Platform. In: Proc. ACM Int. workshop on Crowdsourcing for multimedia (CrowdMM), pp. 21–22. doi:10.1145/2506364.2506371.

Amazon Mechanical Turk Human Intelligence through an API. [Online]. Available: http://mturk.com.

Amsterdamer, Y., Grossman, Y., Milo, T., Senellart, P., 2013. Crowdminer: mining association rules from the crowd. Proceedings of the VLDB Endowment 6 (12), 1250–1253. doi:10.14778/2536274.2536288.

Anderson, J., Soden, R., Keegan, B., Palen, L., Kenneth, M., 2018. The crowd is the territory : assessing quality in peer-Produced spatial data during disasters. Int. J. Human Comput. Interact. (IJHCI) 34 (4), 295–310. doi:10.1080/10447318.2018.1427828.

Aris, H., 2017. Current State of Crowdsourcing Taxonomy Research: A Systematic Review. In: Proc. Int. Conf. on Comput. and Informatics (ICOCI), pp. 392–399.

Aris, H., Din, M.M., 2016. Crowdsourcing evolution: Towards a taxonomy of crowdsourcing initiatives. In: Proc. IEEE Int. Conf. on Perv. Comput. and Comm. (PerCom), pp. 1–6. doi:10.1109/PERCOMW.2016.7457122.

Artikis, A., Weidlich, M., Schnitzler, F., Boutsis, I., Liebig, T., Piatkowski, N., Bockermann, C., Morik, K., Kalogeraki, V., Marecek, J., Gal, A., Mannor, S., Kinane, D., Gunopulos, D., 2014. Heterogeneous Stream Processing and Crowdsourcing for Urban Traffic Management. In: Proc. Int. Conf. on Extend. Database Technol. (EDBT), pp. 712–723. doi:10.5441/002/edbt.2014.77.

Assadi, S., Hsu, J., Jabbari, S., 2015. Online Assignment of Heterogeneous Tasks in Crowdsourcing Markets. In: Proc. AAAI Conf. Human Comput. and Crowdsourc. (HCOMP), pp. 12–21.

Bauer, J., Franke, N., Tuertscher, P., 2016. Intellectual property norms in online communities: how user-organized intellectual property regulation supports innovation. Inf. Syst. Research (ISR) 27 (4), 724–750. doi:10.1287/isre.2016.0649.

Bernstein, M.S., Little, G., Miller, R.C., Hartmann, B., Ackerman, M.S., Karger, D.R., Crowell, D., Panovich, K., 2010. Soylent: a word processor with a crowd inside. Commun. ACM 58 (8), 85–94. doi:10.1145/1866029.1866078.

Bernstein, M.S.M., Brandt, J., Miller, R.C., Karger, D.R., 2011. Crowds in Two Seconds: Enabling Realtime Crowd-powered Interfaces. In: Proc. ACM Symp. on User Inter. Softw. and Technol. (UIST), pp. 33–42. doi:10.1145/2047196.2047201.

Bigham, J.P., Jayant, C., Ji, H., Little, G., Miller, A., Miller, R.C., Miller, R., Tatarowicz, A., White, B., White, S., Yeh, T., 2010. VizWiz : Nearly Real-time Answers to Visual Questions. In: Proc. ACM Symp. on User Inter. Softw. and Technol. (UIST), pp. 333–342.

Boim, R., Greenshpan, O., Milo, T., Novgorodov, S., Polyzotis, N., Tan, W.C., 2012. Asking the right questions in crowd data sourcing. In: Proc. IEEE Int. Conf. on Data Eng. (ICDE), pp. 1261–1264. doi:10.1109/ICDE.2012.122.

Bonabeau, E., 2009. Decisions 2.0: the power of collective intelligence. MIT Sloan Manag. Rev. 50 (2), 45–52.

Boutsis, I., Kalogeraki, V., 2014. On task assignment for real-time reliable crowdsourcing. In: Proc. IEEE Int. Conf. Dist. Comput. Syst. (ICDCS), pp. 1–10. doi:10.1109/ICDCS.2014.9.

Bozat, Z.A., 2017. Crowdsourcing as an open innovation tool for entrepreneurship. Int. J. of Eco. and Manag. Eng. 11 (5), 883–887.

Brabham, D.C., 2008. Crowdsourcing as a model for problem solving: an introduction and cases. Convergence: The Int. J. of Res. into New Media Technol. 14 (1), 75–90. doi:10.1177/1354856507084420.

Brabham, D.C., 2009. Crowdsourcing the public participation process for planning projects. Plan. Th. 8 (3), 242–262.

Brabham, D.C., 2010. Moving the crowd at threadless: motivations for participation in a crowdsourcing application. Inf. Commun. and Society (iCS) 13 (8), 1122–1145. doi:10.1080/13691181003624090.

Brabham, D.C., 2013. "Crowdsourcing". Wiley Online Library.

Brabham, D.C., 2013. Using crowdsourcing in government. IBM.

Breazeal, C., DePalma, N., Orkin, J., Chernova, S., Jung, M., 2013. Crowdsourcing human-Robot interaction: new methods and system evaluation in a public environment. J. of Human-Robot Interaction (JHRI) 2 (1), 82–111. doi:10.5898/JHRI.2.1.Breazeal.

Carusi, C., Bianchi, G., Bracciale, L., 2018. An online approach for joint task assignment and worker evaluation in crowd-sourcing. Pervasive and Mobile Computing (PMC) 50, 94–113. doi:10.1016/j.pmcj.2018.08.004.

Carvalho, A., Dimitrov, S., Larson, K., 2016. How many crowdsourced workers should a requester hire? Ann. Math. Artif. Intell. 78 (1), 45–72. doi:10.1007/s10472-015-9492-4.

Castano, S., Ferrara, A., Montanelli, S., 2016. Designing crowdsourcing tasks with consensus constraints. In: Proc. IEEE Int. Conf. on Collab. Technol. and Syst. (CTS), pp. 97–103. doi:10.1109/CTS.2016.33.

Catallo, I., Martinenghi, D., 2017. The Dimensions of Crowdsourcing Task Design. In: Proc. 17th Int. Conf. Web. Engg. (ICWE), pp. 394–402. doi:10.1007/978-3-319-60131-1.

Chen, J.J., Menezes, N.J., Bradley, A.D., 2011. Opportunities for Crowdsourcing Research on Amazon Mechanical Turk. In: Proc. Workshop on Crowdsourc. and Human Comput. (CHI), pp. 3–6. doi:10.1145/1357054.1357127.

Chen, J.J., Yu, F., 2016. Gnafuy: A framework for ubiquitous mobile computation. In: Proc. IEEE Int. Conf. on Cloud Comput. (CLOUD), pp. 810–813. doi:10.1109/CLOUD.2016.111.

Chen, S., Li, M., Ren, K., Qiao, C., 2015. Crowd Map: Accurate Reconstruction of Indoor Floor Plans from Crowdsourced Sensor-Rich Videos. In: Proc. IEEE Int. Conf. Dist. Comput. Syst. (ICDCS), pp. 1–10. doi:10.1109/ICDCS.2015.9.

Chittilappilly, A.I., Chen, L., Amer-Yahia, S., 2016. A survey of general-Purpose crowdsourcing techniques. IEEE Trans. Knowl. Data Eng. (TKDE) 28 (9), 2246–2266. doi:10.1109/TKDE.2016.2555805.

Chiu, C.M., Liang, T.P., Turban, E., 2014. What can crowdsourcing do for decision support? Decision Support Syst. (DSS) 65, 40–49. doi:10.1016/j.dss.2014.05.010.

Chowdhury, S.A., Calvo, M., Ghosh, A., Stepanov, E.A., Bayer, A.O., Riccardi, G., García, F., Sanchis, E., 2015. Selection and aggregation techniques for crowdsourced semantic annotation task. In: Proc. Conf. of Int. Speech Comm. Assoc. (ISCA), pp. 2779–2783.

Chowdhury, S.N., Dhawan, S., Agnihotri, A., 2016. Crowd-Sourcing for smart cities. In: Proc. IEEE Int. Conf. on Recent Trends in Elect. Inf. and Commun. Technol. (RTEICT), pp. 360–365. doi:10.1109/RTEICT.2016.7807842.

Chrum, A. "The Long History of Crowdsourcing - and Why You're Just Now Hearing About It", 2015.

Cooper, S., Khatib, F., Makedon, I., Lu, H., Barbero, J., Baker, D., Fogarty, J., Popović, Z., players, F., 2011. Analysis of Social Gameplay Macros in the Foldit Cookbook. In: Proc. ACM Int. Conf. on Foundations of Digital Games (FDG), pp. 9–14. doi:10.1145/2159365.2159367.

Corney, J.R., Torres-Sánchez, C., Jagadeesan, A.P., Yan, X.T., Regli, W.C., Medellin, H., 2010. Putting the crowd to work in a knowledge-based factory. Adv. Eng. Informatics 24 (3), 243–250. doi:10.1016/j.aei.2010.05.011.

Cox, L.P., 2011. Truth in crowdsourcing. IEEE Secur. Privacy 9 (5), 74–76. doi:10.1109/MSP.2011.145.

Crowdspring Design Done Better[Online]. Available: http://crowdspring.com/.

Dai, P., Lin, C.H., Mausam, Weld, D.S., 2013. POMDP-Based control of workflows for crowdsourcing (SINGLE CHOICE MODEL). Artif Intell 202, 52–85. doi:10.1016/j.artint.2013.06.002.

Daltayanni, M., de Alfaro, L., Papadimitriou, P., 2015. WorkerRank: Using Employer Implicit Judgements To Infer Worker Reputation. In: Proc. ACM Int. Conf. on Web Search and Data Mining (WSDM), pp. 263–272. doi:10.1145/2684822.2685286.

Daniel, F., Kucherbaev, P., Cappiello, C., Benatallah, B., Allahbakhsh, M., 2018. Quality control in crowdsourcing: A Survey of quality attributes, assessment techniques and assurance actions. ACM Comput. Surv. 51 (1), 1–40. doi:10.1145/3148148.

Dawid, A.P., Skene, A.M., 1979. Maximum likelihood estimation of observer error-Rates using the EM algorithm. J. of the Royal Statistical Society 28 (1), 20–28. doi:10.2307/2346806.

Design and Build High-Quality Software with Crowdsourcing [Online]. Available: http://www.topcoder.com/.

Difallah, D.E., Catasta, M., Demartini, G., Cudré-Mauroux, P., 2014. Scaling-up the Crowd: Micro-Task Pricing Schemes for Worker Retention and Latency Improvement. In: Proc. AAAI Conf. Human Comput. and Crowdsourc. (HCOMP), pp. 50–58. doi:10.13140/2.1.3692.8649.

Doan, A., Ramakrishnan, R., Halevy, A.Y., 2011. Crowdsourcing systems on the world-Wide web. Commun. ACM 54 (4), 86–96. doi:10.1145/1924421.1924442.

DotSUB The Leading Way to Caption[Online]. Available: https://dotsub.com/.

Dubach, E., Muhdi, L., Stöcklin, D., Michahelles, F., 2011. Crowdsourcing for Kiosk of the Future A Retail Store Case Study. In: Proc. Americas Conf. Inf. Syst. (AMCIS), pp. 1–9.

Duolingo A Freemium Language-learning Platform[Online]. Available: http://duolingo.cn/.

Durward, D., Blohm, I., Leimeister, J.M., 2016. Is There PAPA in Crowd Work? A Literature Review on Ethical Dimensions in Crowdsourcing. In: Proc. IEEE Int. Conf. on Ubi. Intell. & Comput. (UIC), pp. 823–832. 10.1109

Eickhoff, C., de Vries, A.P., 2013. Increasing cheat robustness of crowdsourcing tasks. Inf. Retrieval (IR) 16 (2), 121–137. doi:10.1007/s10791-011-9181-9.

Estellés-Arolas, E., González-Ladrón-De-Guevara, F., 2012. Towards an integrated crowdsourcing definition. J. Inf. Sci. 38 (2), 189–200. doi:10.1177/016555150000000.

Felstiner, A., 2011. Working the crowd : employment and labor law in the crowdsourcing industry. Berkeley J. of Emp. & Labor Law (BJELL) 32 (1), 143–204. doi:10.15779/Z38Z92X.

FigureEight AI Platform for Data Science and Machine Learning[Online]. Available: https://www.figure-eight.com/.

Finnerty, A., Kucherbaev, P., Tranquillini, S., Convertino, G., 2013. Keep it Simple: Reward and Task Design in Crowdsourcing. In: Proc. ACM Conf. Italian Chapter SIGCHI (CHItaly), p. 14. doi:10.1145/2499149.2499168.

Fortune Data Store [Online]. Available:http://fortune.com/fortune500/.

From "TOYODA" to "TOYOTA". [Online]. Available: http://toyota-global.com/showroom/emblem/history/.

Gadiraju, U., Kawase, R., Dietze, S., 2015. Understanding Malicious Behavior in Crowdsourcing Platforms : The Case of Online Surveys. In: Proc. ACM Int. Conf. Human Factors in Comput. Syst. (CHI), pp. 1631–1640.

Garcia-Molina, H., Joglekar, M., Marcus, A., Parameswaran, A., Verroios, V., 2016. Challenges in data crowdsourcing. IEEE Trans. Knowl. Data Eng. (TKDE) 28 (4), 901–911. doi:10.1109/TKDE.2016.2518669.

Georgescu, M., Zhu, X., 2014. Aggregation of Crowdsourced Labels Based on Worker History. In: Proc. ACM Int. Conf. on Web Intell. Mining and Semantics (WIMS), pp. 1–11. doi:10.1145/2611040.2611074.

Ghezzi, A., Gabelloni, D., Martini, A., Natalicchio, A., 2017. Crowdsourcing: A Review and suggestions for future research. Int. J. of Manag. Reviews (IJMR) 20, 343–363. doi:10.1111/ijmr.12135.

Gneezy, U., Rustichini, A., 2000. Pay enough or Don't pay at all. The Quarterly J. of Economics 115 (3), 791–810.

Good, B.M., Su, A.I., 2013. Crowdsourcing for bioinformatics. Bioinformatics 29 (16), 1925–1933. doi:10.1093/bioinformatics/btt333.

Gurari, D., Sameki, M., Betke, M., 2016. Investigating the Influence of Data Familiarity to Improve the Design of a Crowdsourcing Image Annotation System. In: In 4th AAAI Conf. Human Comput. and Crowdsourc. (HCOMP), pp. 59–68.

Haas, D., Ansel, J., Gu, L., Marcus, A., 2015. Argonaut : macrotask crowdsourcing for complex data processing. Proc. of the VLDB Endow. (PVLDB) 8 (12), 1642–1653. doi:10.14778/2824032.2824062.

Hackman, J., Oldham, G.R., 1976. Motivation through the design of work: test of a theory. Organ. Behav. Hum. Perform. 16 (2), 250–279. doi:10.1016/0030-5073(76)90016-7.

Hamari, J., Koivisto, J., Sarsa, H., 2014. Does gamification work? - A literature review of empirical studies on gamification. In: Proc. IEEE Int. Conf. on Syst. Sci. (HICSS), pp. 3025–3034. doi:10.1109/HICSS.2014.377 arXiv:1011.1669v3.

Heimerl, K., Gawalt, B., Chen, K., Parikh, T., Hartmann, B., 2012. CommunitySourcing: Engaging Local Crowds to Perform Expert Work Via Physical Kiosks. In: Proc. ACM Int. Conf. Human Factors in Comput. Syst. (CHI), pp. 1539–1548. doi:10.1145/2207676.2208619.

Hetmank, L., 2013. Components and Functions of Crowdsourcing Systems A Systematic Literature Review. In: Proc. Int. Conf. on Wirtschaftsinformatik, pp. 55–69. doi:10.1145/1400214.1400244 arXiv:1011.1669v3.

Hirth, M., HoBfeld, T., Tran-Gia, P., 2011. Cost-optimal validation mechanisms and cheat-detection for crowdsourcing platforms. In: Proc. IEEE Int. Conf. Innov. Mob. and Inter. Serv. in Ubi. Comput. (IMIS), pp. 316–321. doi:10.1109/IMIS.2011.91.

Hirth, M., Hoßfeld, T., Tran-Gia, P., 2013. Analyzing costs and accuracy of validation mechanisms for crowdsourcing platforms. Int. J. Math. and Comput. Model. (MCM) 57 (11–12), 2918–2932. doi:10.1016/j.mcm.2012.01.006.

Hopkins, R., 2011. What Is Crowdsourcing?, Kogan Page, London. In: Paul Sloane (Ed.), A Guide to Open Innovation and Crowdsourcing: Advice from Leading Experts, pp. 60–64. doi:10.1016/j.bmcl.2014.08.032.

Hossain, M., Kauranen, I., 2015. Crowdsourcing: a comprehensive literature review. Strategic Outsourcing: An Int. J. 8 (1), 2–22. doi:10.1108/SO-12-2014-0029.

Hosseini, M., Shahri, A., Phalp, K., Taylor, J., Ali, R., 2015. Crowdsourcing: a taxonomy and systematic mapping study. Computer Science Review 17, 43–69. doi:10.1016/j.cosrev.2015.05.001.

Howe, J., a. "Crowdsourcing: A Definition", 2006.

Howe, J., b. "The Rise of Crowdsourcing", 2006.

Howe, J., 2008. "Crowdsourcing: Why the power of the crowd is driving the future of business", 1st Crown Publishing, NY, USA.

Huang, J., Sun, X., Zhou, Y., Sun, H., 2017. A team formation model with personnel work hours and project workload quantified. Comput. J. 60 (9), 1382–1394. doi:10.1093/comjnl/bxx009.

Hung, N.Q.V., Thang, D.C., Weidlich, M., Aberer, K., 2015. Minimizing Efforts in Validating Crowd Answers. In: Proc. ACM Int. Conf. Manag. of Data (SIGMOD), pp. 999–1014. doi:10.1145/2723372.2723731.

Ikeda, K., Morishima, A., Rahman, H., Roy, S.B., Thirumuruganathan, S., Amer-Yahia, S., Das, G., 2016. Collaborative crowdsourcing with crowd4u. Proc. of the VLDB Endow. (PVLDB) 9 (13), 1497–1500. doi:10.14778/3007263.3007293.

Image Labeler - Crowdsource - Google [Online]. Available:https://crowdsource.google.com/imagelabeler/category.

Inel, O., Khamkham, K., Cristea, T., Dumitrache, A., Rutjes, A., van der Ploeg, J., Romaszko, L., Aroyo, L., Sips, R.-J., 2014. CrowdTruth: Machine-Human Computation Framework for Harnessing Disagreement in Gathering Annotated Data. In: Proc. Int. Semantic Web Conf. (ISCW), pp. 486–504. doi:10.1007/978-3-319-11915-1_31.

InnoCentive Innovate with InnoCentive[Online]. Available: http://innocentive.com.

Irani, L.C., Silberman, M.S., 2013. Turkopticon: Interrupting worker invisibility in amazon mechanical turk. In: Proc. ACM Int. Conf. Human Factors in Comput. Syst. (CHI), pp. 611–620. doi:10.1145/2470654.2470742.

iStockPhoto International Micro Stock Photography Provider[Online]. Available: http://istockphoto.com/.

Iyyer, M., Yih, W.-t., Chang, M.-W., 2016. Answering complicated question intents expressed in decomposed question sequences. CoRR abs/1611.0, 1–9.

Iyyer, M., Yih, W.-T., Chang, M.-W., 2017. Search-based Neural Structured Learning for Sequential Question Answering. In: Proc. Meeting of Assoc. for Comput. Ling. (ACL), pp. 1821–1831. doi:10.18653/v1/P17-1167.

Jiang, H., Matsubara, S., 2014. Efficient Task Decomposition in Crowdsourcing. In: Proc. Int. Conf. on Principles and Pract. of Multi-Agent Syst. (PRIMA), pp. 65–73. doi:10.1007/978-3-319-13191-7_6.

Jiang, L., Wagner, C., Nardi, B., 2015. Not Just in it for the Money: A Qualitative Investigation of Workers' Perceived Benefits of Micro-task Crowdsourcing. In: Proc. IEEE Int. Conf. on Syst. Sci. (HICSS), pp. 773–782. doi:10.1109/HICSS.2015.98.

Jin, Y., Carman, M., Zhu, Y., Xiang, Y., 2018. A Technical Survey on Statistical Modelling and Design Methods for Crowdsourcing Quality Control. CoRR abs/1812.0, 1–50. arXiv:1812.02736v1

Jun, Y., Shaoping, K., 2017. Research on the Reputation Model Based on Active Degree of Crowd-sourcing Workers. In: Proc. Int. Multiconf. Eng. and Comput. Scientists (IMECS), pp. 1–5.

Karger, D.R., Oh, S., Shah, D., 2011. Iterative Learning for Reliable Crowdsourcing Systems. In: Proc. Conf. on Neural Inf. Process. Syst. (NIPS), pp. 1953–1961 arXiv:1110.3564v2.

Kärkkäinen, H., Jussila, J., Multasuo, J., 2012. Can crowdsourcing really be used in B2B innovation? In: Proc. ACM Int. Conf. on Media of the Future (MindTrek), pp. 134–141. doi:10.1145/2393132.2393159.

Kaufmann, N., Schulze, T., Veit, D., 2011. More than fun and money. Worker Motivation in Crowdsourcing A Study on Mechanical Turk. In: Proc. Americas Conf. Inf. Syst. (AMCIS), pp. 1–11. doi:10.1145/1979742.1979593.

Kazai, G., 2010. An exploration of the influence that task parameters have on the performance of crowds. CrowdConf, Oct.

Khan Khattak, F., Salleb-Aouissi, A., 2011. Quality Control of Crowd Labeling through Expert Evaluation. In: NIPS 2nd Workshop Comput. Social Sci. and the Wisdom of Crowds, pp. 1–5.

Kittur, A., Chi, E.H., Suh, B., 2008. Crowdsourcing User Studies With Mechanical Turk. In: Proc. ACM Int. Conf. Human Factors in Comput. Syst. (CHI), pp. 453–456. doi:10.1145/1357054.1357127.

Kittur, A., Khamkar, S., André, P., Kraut, R.E., 2012. CrowdWeaver : Visually Managing Complex Crowd Work. In: Proc. ACM Conf. Computer Supp. Coop. Work (CSCW), pp. 1033–1036. doi:10.1145/2145204.2145357.

Kittur, A., Smus, B., Kraut, R., 2011. CrowdForge: Crowdsourcing Complex Work. In: Proc. ACM Symp. on User Inter. Softw. and Technol. (UIST), pp. 43–52. doi:10.1145/1979742.1979902.

Kocsis, D., de Vreede, G.-J., 2016. Towards a Taxonomy of Ethical Considerations in Crowdsourcing. In: Proc. Americas Conf. Inf. Syst. (AMCIS), pp. 1–10.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. Adv. Neural Inf. Proc. Syst. 1097–1105. https://doi.org/10.1016/j.protcy.2014.09.007.

Kucherbaev, P., Daniel, F., Tranquillini, S., Marchese, M., 2016. Crowdsourcing processes: A Survey of approaches and opportunities. IEEE Internet Comput. 20 (2), 50–56. doi:10.1109/MIC.2015.96.

Kulkarni, A., Can, M., Hartmann, B., 2011. Turkomatic: Automatic, Recursive Task and Workflow Design for Mechanical Turk. In: Proceedings of the annual ACM conference extended abstracts on Human factors in computing systems, pp. 2053–2058. doi:10.1145/1979742.1979865.

Kulkarni, A., Can, M., Hartmann, B., 2012. Collaboratively crowdsourcing workflows with turkomatic. In: Proc. ACM Conf. Computer Supp. Coop. Work (CSCW), pp. 1003–1012. doi:10.1145/2145204.2145354.

Kuncheva, L.I., Whitaker, C.J., Shipp, C.A., Duin, R.P.W., 2003. Limits on the majority vote accuracy in classifier fusion. Pattern Analysis and Appl. 6 (1), 22–31. doi:10.1007/s10044-002-0173-7.

Land, K., Slosar, A., Lintott, C., Andreescu, D., Bamford, S., Murray, P., Nichol, R., Raddick, M.J., Schawinski, K., Szalay, A., Thomas, D., Vandenberg, J., 2008. Galaxy zoo: the large-scale spin statistics of spiral galaxies in the sloan digital sky survey. Mon. Not. R. Astron. Soc. 388 (4), 1686–1692. doi:10.1111/j.1365-2966.2008.13490.x.

Lappas, T., Liu, K., Terzi, E., 2009. Finding a team of experts in social networks. In: Proc. ACM Int. Conf. Knowl. Disc. and Data Min. (KDD), pp. 467–476. doi:10.1145/1557019.1557074.

Lasecki, W., Miller, C., Sadilek, A., Abumoussa, A., Borrello, D., Kushalnagar, R., Bigham, J., 2012. Real-time captioning by groups of non-experts. In: Proc. ACM Symp. on User Inter. Softw. and Technol. (UIST), pp. 23–34. doi:10.1145/2380116.2380122.

LaToza, T.D., Ben Towne, W., Adriano, C.M., Van Der Hoek, A., 2014. Microtask programming: building software with a crowd. In: Proc. ACM Symp. on User Inter. Softw. and Technol. (UIST), pp. 43–54. doi:10.1145/2642918.2647349.

Lee, K., Caverlee, J., Webb, S., 2010. The social honeypot project: protecting online communities from spammers. In: Proc. ACM Int. Conf. on World Wide Web (WWW), pp. 1139–1140. doi:10.1145/1772690.1772843.

Li, G., Wang, J., Zheng, Y., Franklin, M., 2016. Crowdsourced data management: A Survey. IEEE Trans. Knowl. Data Eng. (TKDE) 28 (9), 2296–2319. doi:10.1109/ICDE.2017.26.

Li, G., Zheng, Y., Fan, J., Wang, J., Cheng, R., 2017. Crowdsourced Data Management: Overview and Challenges. In: Proc. ACM Int. Conf. Manag. of Data (SIGMOD), pp. 1711–1716. doi:10.1145/3035918.3054776.

Li, Q., Vempaty, A., Varshney, L.R., Varshney, P.K., 2017. Multi-Object classification via crowdsourcing with a reject option. IEEE Trans. Signal Processing 65 (4), 1068–1081. doi:10.1109/TSP.2016.2630038.

Liang, H., Wang, M.M., Wang, J.J., Xue, Y., 2018. How intrinsic motivation and extrinsic incentives affect task effort in crowdsourcing contests: a mediated moderation model. Comput. Human Behav. 81, 168–176. doi:10.1016/j.chb.2017.11.040.

Little, G., Chilton, L.B., Goldman, M., Miller, R.C., 2010. TurKit : Human Computation Algorithms on MTurk. In: Proc. ACM Symp. on User Inter. Softw. and Technol. (UIST), pp. 57–66. doi:10.1145/1866029.1866040.

Little, G., Chilton, L.B., Miller, R.C., Goldman, M., 2009. TurKit : Tools for iterative tasks on mechanical Turk. In: Proc. ACM Workshop on Human Comput. (SIGKDD), pp. 29–30.

Littlestone, N., Warmuth, M., 1994. The weighted majority algorithm. Inf. Comput. 108 (2), 212–261. doi:10.1006/inco.1994.1009.

Liu, A., Lu, S.C., 2016. A crowdsourcing design framework for concept generation. CIRP Annals - Manufacturing Technol. 65 (1), 177–180. doi:10.1016/j.cirp.2016.04.021.

Liu, Q., Luo, T., Tang, R., Bressan, S., 2018. An efficient and truthful pricing mechanism for team formation in crowdsourcing markets. CoRR abs/1812.0, 567–572. doi:10.1109/ICC.2015.7248382.

Liu, S., Chen, C., Lu, Y., Ouyang, F., Wang, B., 2018. An interactive method to improve crowdsourced annotations. IEEE Trans. Vis. Comput. Graph. (TVCG) 25 (1), 235–245. doi:10.1109/TVCG.2018.2864843.

Liu, X., Lu, M., Ooi, B.C., Shen, Y., Wu, S., Zhang, M., 2012. CDAS: A Crowdsourcing Data analytics system. PVLDB 5 (10), 1040–1051. doi:10.14778/2336664.2336676.

Liu, Y., Lehdonvirta, V., Alexandrova, T., Nakajima, T., 2012. Drawing on mobile crowds via social media case ubiask: image based mobile social search across languages. Multimedia Syst. 18 (1), 53–67. doi:10.1007/s00530-011-0242-0.

Liu, Y., Liu, M., 2017. An online learning approach to improving the quality of crowd-Sourcing. IEEE/ACM Trans. Netw. (TON) 25 (4), 2166–2179. doi:10.1145/2745844.2745874.

Luz, N., Silva, N., Novais, P., 2015. A survey of task-oriented crowdsourcing. Artif. Intell. Rev. 44 (2), 187–213. doi:10.1007/s10462-014-9423-5.

Lynch, A. "Crowdsourcing Is Not New - The History of Crowdsourcing (1714 to 2010)", 2010.

Maiolini, R., 2011. Information technology and innovation trends in organizations. Physica-Verlag HD - Springer doi:10.1007/978-3-7908-2632-6.

Mao, K., Capra, L., Harman, M., Jia, Y., 2017. A survey of the use of crowdsourcing in software engineering. J. of Syst. and Softw. (JSS) 126 (4), 57–84. doi:10.1016/j.jss.2016.09.015.

Marcus, A., Karger, D., Madden, S., Miller, R., Oh, S., 2012. Counting with the crowd. Proc. of the VLDB Endow. (PVLDB) 6 (2), 109–120.

Marjanovic, S., Fry, C., Chataway, J., 2012. Crowdsourcing based business models: in search of evidence for innovation 2.0. Science and Public Policy 39 (3), 318–332. doi:10.1093/scipol/scs009.

Mashhadi, A.J., Capra, L., 2011. Quality control for real-time ubiquitous crowdsourcing. In: Proc. ACM Int. workshop on Ubiquitous crowdsouring (UbiCrowd), pp. 5–8. doi:10.1145/2030100.2030103.

Mason, W., Suri, S., 2012. Conducting behavioral research on Amazon's mechanical turk. Behavior Research Methods (BRM) 44 (1), 1–23. doi:10.3758/s13428-011-0124-6.

Mihalcea, R., Chklovski, T., 2004. Building sense tagged corpora with volunteer contributions over the web. Recent Advances in Natural Language Processing III 260, 357.

Miniukovich, A., De Angeli, A., 2015. Visual diversity and user interface quality. In: Proc. ACM British HCI, pp. 101–109. doi:10.1145/2783446.2783580.

Moayedikia, A., Ong, K.L., Boo, Y.L., Yeoh, W.G., 2018. Task assignment in microtask crowdsourcing platforms using learning automata. Eng. Appl. of AI 74, 212–225. doi:10.1016/j.engappai.2018.06.008.

Moayedikia, A., Yeoh, W., Ong, K.L., Boo, Y.L., 2017. Framework and literature analysis for Crowdsourcing's answer aggregation. J. of Comput. Inf. Syst. (JCIS) 0 (0), 1–12. doi:10.1080/08874417.2017.1394171.

Morschheuser, B., Hamari, J., Koivisto, J., 2016. Gamification in crowdsourcing: A review. In: Proc. IEEE Int. Conf. on Syst. Sci. (HICSS), pp. 4375–4384. doi:10.1109/HICSS.2016.543.

Muldoon, C., O'Grady, M.J., O'Hare, G.M.P., 2018. A survey of incentive engineering for crowdsourcing. Knowledge Eng. Review (KER) 33 (e2), 1–19. doi:10.1017/S0269888918000061.

Murano, P., 2018. A new user interface for a text proofreading web portal in a digitization and crowdsourcing context. Int. J. of Web Inf. Syst. (IJWIS) 14 (1), 107–122. doi:10.1108/IJWIS-05-2017-0042.

Murturi, A., Kantarci, B., Oktug, S.F., 2015. A reference model for crowdsourcing as a service. In: Proc. IEEE Int. Conf. on Cloud Netw. (CloudNet), pp. 64–66. doi:10.1109/CloudNet.2015.7335281.

Nakatsu, R.T., Grossman, E.B., Lacovou, C.L., 2014. A taxonomy of crowdsourcing based on task complexity. J. Inf. Sci. (JIS) 40 (6), 823–834. doi:10.1177/1045389X14554132.

Nassar, L., Karray, F., 2019. Overview of the crowdsourcing process. Knowl. and Inf. Systems 60 (1), 1–24. doi:10.1007/s10115-018-1235-5.

Nguyen, Q.V.H., Nguyen, T.T., Lam, N.T., Aberer, K., 2013. BATC - A Benchmark for Aggregation Techniques in Crowdsourcing. In: Proc. ACM Int. Conf. on Res. and Dev. in Inf. Retriev. (SIGIR), pp. 1079–1080. doi:10.1145/2484028.2484199.

Noronha, J., Hysen, E., Zhang, H., Gajos, K.Z., 2011. PlateMate: Crowdsourcing Nutrition Analysis from Food Photographs. In: Proc. ACM User Inter. Softw. and Technol. (UIST), pp. 1–12. doi:10.1145/2047196.2047198.

OpenIDEO Social Impact Powered by Design Thinking[Online]. Available: https://quirky.com/.

Orting, S., Doyle, A., van Hilten, M.H.A., Inel, O., Madan, C.R., Mavridis, P., Spiers, H., Cheplygina, V., 2019. A survey of crowdsourcing in medical image analysis. CoRR abs/1902.0, 1–12.

Parameswaran, A., Teh, M.H., Garcia-molina, H., Widom, J., 2014. DataSift : A Crowd-Powered Search Toolkit. In: Proc. ACM Int. Conf. Manag. of Data (SIGMOD), pp. 885–888. doi:10.1145/2588555.2594510.

Park, H., Widom, J., 2014. Crowdfill: collecting structured data from the crowd. Proc. ACM Int. Conf. Manag. of Data (SIGMOD) 577–588. doi:10.1145/2588555.2610503.

Peng, G., 2011. On the legal issues about crowdsourcing and witkey. In: Proc. IEEE Int. Conf. on Edu., Entertain. and e-Manag. (ICEEE), pp. 293–295.

Pilourdault, J., Amer-yahia, S., Lee, D., Roy, S., 2017. Motivation-Aware Task Assignment in Crowdsourcing. In: Proc. Int. Conf. on Extend. Database Technol. (EDBT), pp. 246–257.

Poonkuzhali, S., Mohana, E., 2016. Crowdsourcing an area to explore. Int. J. Adv. Eng. Technol. (IJAET) VII, 471–477.

Pu, L., Chen, X., Xu, J., Fu, X., 2017. Crowd foraging: a QoS-Oriented self-Organized mobile crowdsourcing framework over opportunistic networks. IEEE J. on Selected Areas in Commun. 35 (4), 848–862. doi:10.1109/JSAC.2017.2679598.

Qin, H., Li, B., Zhang, Y., Wang, J., Lee, S., 2017. Truthful Mechanism for Crowdsourcing Task Assignment. In: Proc. IEEE Int. Conf. on Cloud Comput. (CLOUD), pp. 520–527. doi:10.1109/CLOUD.2017.72.

Quoc Viet Hung, N., Thanh Tam, N., Ngoc Tran, L., Aberer, K., 2013. An Evaluation of Aggregation Techniques in Crowdsourcing. In: Proc. Int. Conf. on Web Inf. Syst. Eng. (WISE), pp. 1–15.

Quirky Quirky Creates Products Invented by Real People like You!. [Online]. Available: https://quirky.com/.

Quora A Place to Share Knowledge and Better Understand the World. [Online]. Available: http://quora.com/.

Rahman, H., Roy, S.B., Thirumuruganathan, S., Amer-Yahia, S., Das, G., 2015. Task assignment optimization in collaborative crowdsourcing. In: Proc. IEEE Int. Conf. on Data Mining (ICDM), pp. 949–954. doi:10.1109/ICDM.2015.119 arXiv:1401.1302v1.

Rahmanian, B., Davis, J.G., 2014. User interface design for crowdsourcing systems. In: Proc. ACM Int. Working Conf. on Adv. Visual Interfaces (AVI), pp. 405–408. doi:10.1145/2598153.2602248.

Raykar, V.C., Yu, S., Zhao, L.H., Jerebko, A., Florin, C., Valadez, G.H., Bogoni, L., Moy, L., 2009. Supervised Learning from Multiple Experts: Whom to trust when everyone lies a bit. In: Proc. ACM Int. Conf. on Mach. Learn. (ICML), pp. 889–896. doi:10.1145/1553374.1553488.

Retelny, D., Robaszkiewicz, S., To, A., Lasecki, W.S., Patel, J., Rahmati, N., Doshi, T., Valentine, M., Bernstein, M.S., 2014. Expert crowdsourcing with flash teams. In: Proc. 27th Annu. ACM Symp. on User Interface Softw. and Technol. (UIST), pp. 75–85. doi:10.1145/2642918.2647409.

Rokicki, M., Zerr, S., Siersdorfer, S., 2015. Groupsourcing: Team Competition Designs for Crowdsourcing. In: Proc. ACM Int. Conf. on World Wide Web (WWW), pp. 906–915. doi:10.1145/2736277.2741097.

Rosen, P.A., 2011. Crowdsourcing lessons for organizations. Journal of Decision Systems (JDS) 20 (3), 309–324. doi:10.3166/jds.20.309-324.

Ross, J., Irani, L., Silberman, M.S., Zaldivar, A., Tomlinson, B., 2010. Who are the Crowdworkers? Shifting Demographics in Mechanical Turk. In: Proc. ACM Int. Conf. Human Factors in Comput. Syst. (CHI), pp. 2863–2872. doi:10.1145/1753846.1753873.

Rouse, A., 2010. A preliminary taxonomy of crowdsourcing. ACIS 2010 Proceedings 76. doi:10.1145/1400214.1400244.

Ryan, R.M., Deci, E.L., 2000. Intrinsic and extrinsic motivations: classic definitions and new directions. Contemporary Edu. Psychology 25 (1), 54–67. doi:10.1006/ceps.1999.1020.

samaSOURCE Training Data Powered by Human Intelligence. [Online]. Available: https://samasource.org/.

Saremi, R.L., Yang, Y., 2015. Dynamic Simulation of Software Workers and Task Completion. In: IEEE/ACM Int. WS CrowdSourc. Softw. Engg. (CSI-SE), pp. 17–23. doi:10.1109/CSI-SE.2015.11.

Sar, A., Tosun, A., Alptekin, G.I., 2019. A systematic literature review on crowdsourcing in software engineering. J. of Syst. and Softw. (JSS) 153, 200–219. doi:10.1016/j.jss.2019.04.027.

Satzger, B., Psaier, H., Schall, D., Dustdara, S., 2013. Auction-based crowdsourcing supporting skill management. Inf. Syst. 38 (4), 547–560. doi:10.1016/j.is.2012.09.003.

Saxton, G.D., Oh, O., Kishore, R., 2013. Rules of crowdsourcing: models, issues, and systems of control. IS Manag. 30 (1), 2–20. doi:10.1080/10580530.2013.739883.

Schenk, E., Guittard, C., 2009. Crowdsourcing : what can be outsourced to the crowd, and why ? Innovation (January) 1–29. doi:10.3917/jie.007.0093.

Schenk, E., Guittard, C., 2011. Towards a characterization of crowdsourcing practices. J. Innov. Eco. & Manag. (JIEM) 1 (7), 93–107. doi:10.3917/jie.007.0093.

Schmitz, H., Lykourentzou, I., 2016. It'S about time: online macrotask sequencing in expert crowdsourcing. CoRR abs/1601.0, 1–24.

Schmitz, H., Lykourentzou, I., 2018. Online sequencing of non-Decomposable macrotasks in expert crowdsourcing. ACM Trans. Social Computing 1 (1), 1–33. doi:10.1145/3140459.

Seaborn, K., Fels, D.I., 2015. Gamification in theory and action: a survey. Int. J. Hum.-Comput. Studies 74, 14–31. doi:10.1016/j.ijhcs.2014.09.006.

Shafiee, S., Shafiee, H., Mortazavi, F., 2016. Security of crowd sourcing. In: Proc. IEEE Int. Conf. on e-Comm. with Focus on e-Tourism (ECDC), pp. 1–6. doi:10.1109/ECDC.2016.7492978.

Sheehan, K.B., 2017. Crowdsourcing research: data collection with Amazon's mechanical turk. Commun. Monogr. 85 (1), 1–17. doi:10.1080/03637751.2017.1342043.

Silberman, M.S., Irani, L., Ross, J., 2010. Ethics and tactics of professional crowdwork. ACM Crossroads 17 (2), 39–43. doi:10.1145/1869086.1869100.

Silva, V., Schneider, D., Oliveira, L., de Souza, J., 2017. Towards a framework for supporting community modeling in crowdsourcing systems. In: Proc. IEEE Int. Conf. Comp. Support. Coop. Work Design (CSCWD), pp. 36–41. doi:10.1109/CSCWD.2017.8066667.

Singla, A., Krause, A., 2013. Truthful incentives in crowdsourcing tasks using regret minimization mechanisms. In: Proc. ACM Int. Conf. on World Wide Web (WWW), pp. 1167–1178. doi:10.1145/2488388.2488490.

Sobel, D., 2007. "Longitude: The true story of a lone genius who solved the greatest scientific problem of his time". Bloomsbury USA.

Squadhelp Find the Perfect Name. [Online]. Available: https://squadhelp.com/.

stackoverflow Learn, Share, Build. [Online]. Available: http://stackoverflow.com/.

Standing, S., Standing, C., 2018. The ethical use of crowdsourcing. Business Ethics: A European Review 27 (1), 72–80. doi:10.1111/beer.12173.

Stol, K.-J., Latoza, T.D., Bird, C., 2017. Crowdsourcing for software engineering. IEEE Softw. 34 (2), 30–36. doi:10.1109/MS.2017.52.

Sun, C., Rampalli, N., Yang, F., Doan, A.H., 2014. Chimera: large-scale classification using machine learning, rules, and crowdsourcing. Proceedings of the VLDB Endowment 7 (13), 1529–1540. doi:10.14778/2733004.2733024.

Surowiecki, J., 2005. "The wisdom of crowds: Why the many are smarter than the few and how collective wisdom shapes business, economies, societies and nations". Anchor Group, NY, USA.

Tarrell, A., Tahmasbi, N., Kocsis, D., Pedersen, J., Tripathi, A., Xiong, J., Oh, O., De Vreede, G.-J., 2013. Crowdsourcing : A Snapshot of Published Research. In: Proc. Americas Conf. Inf. Syst. (AMCIS), pp. 1–14. doi:10.1177/1354856507084420.

Teodoro, R., Ozturk, P., Naaman, M., Mason, W., Lindqvist, J., 2014. The motivations and experiences of the on-demand mobile workforce. In: Proc. ACM Conf. Comput. Supp. Coop. Work (CSCW), pp. 236–247. doi:10.1145/2531602.2531680.

Threadless Our Never-ending, No-themes, No-holds-barred, Open-ended Design Challenge. [Online]. Available: http://threadless.com/.

Thuan, N.H., Antunes, A.P., 2013. Crowdsourcing Design Reference: A Preliminary Model. In: New Zealand Inf. Syst. Doc. Consortium (NZISDC), pp. 1–7.

Thuan, N.H., Antunes, P., Johnstone, D., 2016. Factors influencing the decision to crowdsource: asystematic literature review. Inf. Syst. Frontiers (ISF) 18 (1), 47–68. doi:10.1007/s10796-015-9578-x.

Tian, F., Huang, L., 2018. Market-based incentive mechanism design for crowdsourcing. In: IEEE Int. Conf. on Commun. (ICC). IEEE, pp. 1–6. doi:10.1109/ICC.2018.8422389.

To, W.-M., Lai, L.S.L., 2015. Crowdsourcing in china: opportunities and concerns. IT Pro. 17 (3), 53–59. doi:10.1109/MITP.2015.47.

Tranquillini, S., Daniel, F., Kucherbaev, P., Casati, F., 2015. Modeling, enacting, and integrating custom crowdsourcing processes. ACM Trans. Web (TWEB) 9 (2), 1–43. doi:10.1145/2746353.

Trushkowsky, B., Kraska, T., Franklin, M.J., Sarkar, P., Ramachandran, V., 2015. Crowdsourcing enumeration queries: estimators and interfaces. IEEE Trans. Knowl. Data Eng. (TKDE) 27 (7), 1796–1809. doi:10.1109/TKDE.2014.2339857.

Turker Nation Helping mTurk Workers. [Online]. Available: http://www.turkernation.com/.

Tuite, K., Snavely, N., Tabing, N., 2011. PhotoCity : Training Experts at Large-scale Image Acquisition Through a Competitive Game. In: Proc. ACM Int. Conf. Human Factors in Comput. Syst. (CHI), pp. 1383–1392. doi:10.1145/1978942.1979146.

Upwork Get It Done with a Freelancer. [Online]. Available: http://upwork.com/.

Vesdapunt, N., Bellare, K., Dalvi, N., 2014. Crowdsourcing algorithms for entity resolution. Proc. of the VLDB Endow. (PVLDB) 7 (12), 1071–1082.

Von Ahn, L., Blum, M., Hopper, N.J., Langford, J., 2003. CAPTCHA: Using Hard AI Problems for Security. In: Proc. Int. Conf. on Theory and App. of Crypt. Technol. (EUROCRYPT), pp. 294–311. doi:10.1007/3-540-39200-9.

Vuković, M., 2009. Crowdsourcing for enterprises. In: IEEE 5th World Congress on Services, pp. 686–692. doi:10.1109/SERVICES-I.2009.56.

Vukovic, M., Bartolini, C., 2010. Towards a research agenda for enterprise crowdsourcing. In: Proc. Int. Symp. on Leverag. App (ISoLA), pp. 425–434. doi:10.1007/978-3-642-16558-0_36.

Wazny, K., 2017. "Crowdsourcing" ten years in: a review. J. Glob. Health 7 (2). doi:10.7189/jogh.07.020601.

Weber, I., Robertson, S., 2009. Rethinking the ESP Game. In: Proc. ACM Int. Conf. Human Factors in Comput. Syst. (CHI), pp. 3937–3942.

Weld, D.S., Fogarty, J., Skinner, M., 2008. Intelligence in Wikipedia. In: Proc. AAAI Conf. on Artif. Intell., pp. 1609–1614. Doi:10.1.1.141.6970

Whitehill, J., Ruvolo, P., Wu, T., Bergsma, J., Movellan, J., 2009. Whose Vote Should Count More: Optimal Integration of Labels from Labelers of Unknown Expertise. In: Proc. Conf. on Neural Inf. Process. Syst. (NIPS), pp. 2035–2043.

Whitla, P., 2009. Crowdsourcing and its application in marketing activities. Contem. Manag. Research 5 (1), 15–28. doi:10.1177/1473095209104824.

Wikipedia Online Encyclopedia. [Online]. Available: https://wikipedia.org/.

Wolfson, S.M., Lease, M., 2011. Look before you leap: legal pitfalls of crowdsourcing. Association for Inf. Sci. and Technol. (asis&t) 48 (1), 1–10. doi:10.1002/meet.2011.14504801135.

work & earn or offer a micro job [Online]. Available: http://www.microworkers.com/.

Wu, F., Wilkinson, D.M., Huberman, B.A., 2009. Feedback loops of attention in peer production. In: Proc. IEEE Int. Conf. on Comput. Sci. and Eng. (CSE), pp. 409–415. doi:10.1109/CSE.2009.430.

Wu, S., Wang, X., Wang, S., Zhang, Z., Tung, A.K., 2014. K-Anonymity For crowdsourcing database. IEEE Trans. Knowl. Data Eng. (TKDE) 26 (9), 2207–2221. doi:10.1109/TKDE.2013.93.

Xintong, G., Hongzhi, W., Song, Y., Hong, G., 2014. Brief survey of crowdsourcing for data mining. Expert Syst. Appl. 41 (17), 7987–7994. doi:10.1016/j.eswa.2014.06.044.

Yang, J., Adamic, L.A., Ackerman, M.S., 2008. Crowdsourcing and Knowledge Sharing: Strategic User Behavior on Taskcn. In: Proc. ACM conf. on Electronic Commerce (EC), pp. 246–255. doi:10.1145/1386790.1386829.

Yang, J., Redi, J., Demartini, G., Bozzon, A., 2016. Modeling Task Complexity in Crowdsourcing. In: Proc. AAAI Conf. Human Comput. and Crowdsourc. (HCOMP), pp. 249–258.

Yang, Z., Zhang, Z., Bao, Y., Gan, X., Tian, X., Wang, X., 2016. OnTac: Online task assignment for crowdsourcing. In: Proc. IEEE Int. Conf. on Commun. (ICC), pp. 1–6. doi:10.1109/ICC.2016.7511256.

Yin, X., Chen, Y., Li, B., 2017. Task Assignment with Guaranteed Quality for Crowdsourcing Platforms. In: Proc. IEEE/ACM Int. Symp. on Quality of Service (IWQoS), pp. 1–10.

Yin, X., Liu, W., Wang, Y., Yang, C., Lu, L., 2014. What? how? where? a survey of crowdsourcing. Springer doi:10.1007/978-94-007-7618-0.

Yuen, M.-C., King, I., Leung, K.-S., 2011. A Survey of Crowdsourcing Systems. In: Proc. 3rd IEEE Int. Conf. Social Computing (SocialCom/PASSAT), pp. 766–773.

Zhang, H., Horvitz, E., Miller, R.C., Parkes, D.C., 2011. Crowdsourcing General Computation. In: Proc. ACM Int. Conf. Human Factors in Comput. Syst. (CHI), pp. 1–4.

Zhang, R., Liu, L., Cui, L., He, W., Li, H., 2018. Answer Aggregation of Crowdsourcing Employing an Improved EM-Based Approach. In: Proc. Int. Conf. Algo. and Architect. for Parallel Process. (ICA3PP), pp. 280–290. doi:10.1007/978-3-642-33078-0 arXiv:1011.1669v3.

Zhao, Y., Zhu, Q., 2014. Evaluation on crowdsourcing research: current status and future direction. Inf. Syst. Frontiers 16 (3), 417–434. doi:10.1007/s10796-012-9350-4.

Zheng, H., Li, D., Hou, W., 2011. Task design, motivation, and participation in crowdsourcing contests. Int. J. Electronic Commerce (IJEC) 15 (4), 57–88. doi:10.2753/JEC1086-4415150402.

Zheng, Q., Wang, W., Yang, Y., Pan, M., Shi, X., 2017. Crowdsourcing Complex Task Automatically by Workflow Technology. In: Proc. Int. Workshop on Manag. Inf. Process. and Coop. (MIPaC), pp. 17–30. doi:10.1007/978-981-10-3996-6.

Zheng, Y., Li, G., Li, Y., Shan, C., Cheng, R., 2017. Truth inference in crowdsourcing : is the problem solved ? Proc. of the VLDB Endow. (PVLDB) 10 (5), 541–552.

Zheng, Y., Wang, J., Li, G., Cheng, R., Feng, J., 2015. QASCA: A Quality-Aware task assignment system for crowdsourcing applications. In: Proc. ACM Int. Conf. Manag. of Data (SIGMOD), pp. 1031–1046. doi:10.1145/2723372.2749430.

Zhu, B.B., Yan, J., Bao, G., Yang, M., Xu, N., 2014. Captcha as graphical passwords–A new security primitive based on hard AI problems. IEEE Trans. Inf. Forensics and Security 9 (6), 891–904. doi:10.1109/TIFS.2014.2312547.

Zhuang, H., Parameswaran, A., Roth, D., Han, J., 2015. Debiasing Crowdsourced Batches. In: Proc. ACM Int. Conf. on Knowl. Discovery and Data Mining (KDD), pp. 1593–1602. doi:10.1145/2783258.2783316.

Zhuang, H., Young, J., 2015. Leveraging In-Batch Annotation Bias for Crowdsourced Active Learning. In: Proc. ACM Int. Conf. on Web Search and Data Mining (WSDM), pp. 243–252. doi:10.1145/2684822.2685301.

99designs: Design Makes Anything Possible. [Online].Available: https://99designs.com/.

Shahzad Sarwar Bhatti is a PhD candidate at the Department of Computer Science and Engineering, Shanghai Jiao Tong University, China. He received the B.S. degree in 2000 and the M.S. degree in 2015 from Pakistan. He had been affiliated with M/s PTCL (Pakistan Telecomm: Company Limited) as ICT professional working in various domains. His research interest includes crowdsourcing, spatial crowdsourcing cooperative communication, algorithms, and optimiza tion.

Xiaofeng Gao received the B.S. degree in information and computational science from Nankai University, China, in 2004; the M.S. degree in operations research and control theory from Tsinghua University, China, in 2006; and the Ph.D. degree in comp uter science from the University of Texas at Dallas, USA, in 2010. She is currently P rofessor with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, China. Her research interests include wireless communications, data engineering, and combinatorial optimizations. She has authored more than 100 peer reviewed papers and 7 book chapters in the related area, including well archived international journals such as IEEE TC, IEEE TKDE, IEEE TPDS, TCS, and also in well kn own conference proceedings such as INFOCOM, SIGKDD, ICDCS. She has also served on the editorial board of Discrete Mathematics, Algorithms and Applications, and as the PCs and peer reviewer for a number of international conferences and journals.

Guihai Chen is a distinguished professor of Shanghai Jiao Tong University. He earned B.S. degree in computer software from Nanjing University in 1984, M.E. degree in computer applications from Southeast University in 1987, and Ph.D. degree in computer scie nce from the University of Hong Kong in 1997. He had been invited as a visiting professor by Kyushu Institute of Technology in Japan, University of Queensland in Australia and Wayne State University in USA. He has a wide range of research interests with fo cus on parallel computing, wireless networks, data centers, peer to peer computing, high performance computer architecture and data engineering. He has published more than 350 peer reviewed papers, and more than 200 of them are in well archived interna-tion al journals such as IEEE TPDS, IEEE TC, IEEE TKDE, ACM/IEEE TON and ACM TOSN, and also in well known conference proceedings such as HPCA, MOBIHOC, INFOCOM, ICNP, ICDCS, CoNEXT and AAAI. He has won several best paper awards including ICNP 2015 best paper aw ard. His papers have been cited for more than 10000 times according to Google Scholar. He is a CCF fellow.