

A Survey on Task Assignment in Crowdsourcing

DANULA HETTIACHCHI, VASSILIS KOSTAKOS, and JORGE GONCALVES,

The University of Melbourne, Australia

Quality improvement methods are essential to gathering high-quality crowdsourced data, both for research and industry applications. A popular and broadly applicable method is task assignment that dynamically adjusts crowd workflow parameters. In this survey, we review task assignment methods that address: heterogeneous task assignment, question assignment, and plurality problems in crowdsourcing. We discuss and contrast how these methods estimate worker performance, and highlight potential challenges in their implementation. Finally, we discuss future research directions for task assignment methods, and how crowdsourcing platforms and other stakeholders can benefit from them.

CCS Concepts: • **General and reference** → **Surveys and overviews**; • **Information systems** → **Crowdsourcing**; • **Human-centered computing** → *Computer supported cooperative work*;

Additional Key Words and Phrases: Crowdsourcing, data quality, heterogeneous task assignment, plurality problem, question assignment, worker attributes

ACM Reference format:

Danula Hettiachchi, Vassilis Kostakos, and Jorge Goncalves. 2022. A Survey on Task Assignment in Crowdsourcing. *ACM Comput. Surv.* 55, 3, Article 49 (February 2022), 35 pages.

<https://doi.org/10.1145/3494522>

1 INTRODUCTION

Crowdsourcing is the process of gathering information or input of a task from a large number of individuals, typically via the Internet [79]. Crowdsourcing allows task requesters to access a large workforce with diverse skills and capabilities cost-effectively and efficiently compared to hiring experts or dedicated workers [128]. Due to this, crowdsourcing has gained widespread popularity and has also become a critical step in harnessing training data for various machine learning models [157].

As crowdsourced input originates from a multitude of workers where task requesters have limited visibility of their background information or credentials, ensuring high-quality contributions has been a significant research challenge. The literature proposes different quality assurance methods, including training workers [36], providing feedback [37, 57], improving task design [35, 49], implementing task workflows [110], aggregating responses [58, 167], and detecting outliers [78, 88]. Among such methods, matching workers with compatible tasks or “task assignment” has emerged as an important mechanism that can increase the quality of the contributed data [101].

Authors' address: D. Hettiachchi, V. Kostakos, and J. Goncalves, Danula Hettiachchi, School of Computing Technologies, RMIT University, VIC, 3000, Australia; emails: danulah@acm.org, jorge.goncalves@unimelb.edu.au, vassilis.kostakos@unimelb.edu.au.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2022 Copyright held by the owner/author(s). Publication rights licensed to ACM.

0360-0300/2022/02-ART49 \$15.00

<https://doi.org/10.1145/3494522>

While there exist several other surveys related to data quality in crowdsourcing [23, 108, 157, 164], none of them extensively review assignment methods. Our review provides an overview of data quality improvement approaches in crowdsourcing, organised under pre-execution, online, and post-processing methods. Then, we dive deep into task assignment approaches. Particularly, we discuss different methods of modelling and estimating worker performance that takes place prior to the task assignment step. We also distinguish question assignment (match individual questions within the task based on factors like question difficulty, worker quality and current answer confidence) from heterogeneous task assignment where we match workers to specific types of tasks (e.g., image classification, sentiment analysis).

Optimum task assignment is a challenging endeavour due to variations in crowd tasks, the inconsistencies with the availability and the diversity of the worker population. Therefore, researchers present various methods that utilise historic worker data [123], current answer distribution [43, 98], gold-standard questions (questions with known answers) [84], worker attributes [52, 95], and behavioural data [65, 142]. Our review sheds light on how these different methods perform under different scenarios. Furthermore, we discuss broader challenges and limitations of assignment approaches and present future directions for research on task assignment.

Overall, our survey makes the following contributions:

- We provide a detailed overview on existing crowdsourcing data quality improvement techniques that aim to match workers with compatible tasks and questions.
- We identify and review specific methods that address task assignment, question assignment, and plurality problems.
- We discuss challenges in employing different worker performance estimation and assignment methods in a crowdsourcing platform.

1.1 Outline of the Survey

Section 2 describes the method followed in selecting the literature included in this survey. Section 3 briefly reviews data quality improvement methods in crowdsourcing, and Section 4 defines the four task assignment problems that we discuss in the survey. Section 5 elaborates on worker performance modelling and estimation methods, which are two critical steps of task assignment. Then, Section 6 summarises task assignment approaches, including heterogeneous task assignment, question assignment, the plurality problem, and budget allocation methods. Section 7 provides an overview of existing crowdsourcing platforms and their available task assignment methods. Finally, Sections 8 and 9 provide future directions on data quality research in crowdsourcing and concluding remarks of our survey.

2 LITERATURE SELECTION

2.1 Background and Motivation

We note several related surveys that capture different elements of crowdsourcing. Daniel et al. [23] look at overarching quality enhancement mechanisms in crowdsourcing. Their survey organises literature under three segments: quality model, which describes different quality dimensions, quality assessment methods, and quality assurance actions. While Daniel et al. [23] summarise task assignment methods, they are not analysed in detail due to the broader scope of their survey.

Zheng et al. [167] examine 17 truth inference techniques such as majority vote, Zencrowd [26], and Minimax [169]. The survey also presents an evaluation of different methods using five real work datasets. The primary focus of our survey lies outside truth inference methods. However, we provide a summary of truth inference methods in Section 3.3 under post-processing data quality improvement methods.

Table 1. Research Questions Examined in the Survey

Research Question	Description
What are the different ways of matching workers with tasks in crowdsourcing and specific methods proposed to achieve them?	Our survey differentiates task assignment from generic data quality improvement, identifies and defines types of task assignment problems, and provides a detailed review of proposed approaches.
How do we estimate and model worker performance for task assignment?	Performance estimation is an essential step in crowdsourcing task assignment. By dissecting proposed methods into estimation and assignment steps, we provide a detailed outlook of different estimation and modelling methods that can also promote the reuse of components in future implementations.
What are the challenges and limitations of task assignment methods and their availability in existing crowdsourcing platforms?	While many data quality improvement methods have been proposed in the literature, not many of them have been widely adopted in commercial crowdsourcing platforms. We review factors that limit their practical uptake and detail specific task assignment methods available in these platforms.

Li et al. [108] surveys crowdsourced data management with an emphasis on different crowd data manipulation operations such as selection, collection, and join. Their survey organises prior work under quality, cost, and latency control methods. Vaughan [157] also present a comprehensive review on how crowdsourcing methods can benefit machine learning research.

Overall, in contrast to prior literature reviews, our survey sheds light on the task assignment problem in crowdsourcing and discusses related assignment-based quality improvement methods. In particular, our survey examines the research questions outlined in Table 1.

2.2 Literature Selection

We conducted an extensive literature search on the ACM Digital Library using a query that includes keywords “task assignment,” “task routing,” or “data quality” and “crowd*” in the Abstract. We included articles published from 2010 and retrieved 747 records. We reduced the resulting set of papers by limiting to publications from a list of conferences and journals that, to the best of our knowledge, publish work on crowdsourcing and related topics. Selected conferences were AAAI, AAMAS, CHI, CIKM, CSCW, ESEM, HCOMP, HT, ICDE, ICML, IUI, JCDL, KDD, SIGIR, SIGMOD, UbiComp, UIST, WI, WSDM, and WWW. Selected journals were PACM IMWUT, PACM HCI, TKDE, TSC, and VLDB. We also excluded workshops, demo papers, posters, extended abstracts, and so on. Literature from specific venues that are not included in the ACM Digital Library (e.g., HCOMP) was manually screened and added to our dataset. Then, we carefully inspected the remaining papers and filtered out papers that were deemed to not be relevant. Furthermore, the survey also includes several additional papers hand-picked by the authors due to their relevance to the topic.

2.3 Scope

Crowdsourcing extends beyond traditional online crowdsourcing using desktop or laptop computers. Other general types that can overlap include mobile crowdsourcing [127] (e.g., smartphones, tablets), situated crowdsourcing [53, 59, 77] (e.g., public displays), spatial crowdsourcing [56, 153] (e.g., workers attempt location-based tasks including physical tasks), and crowdsensing [62] (e.g., workers passively contribute sensor data from mobile devices). Task assignment in

crowdsourcing has also been investigated based on such domains. However, due to wide variations in techniques used in these different settings, we limit our scope to online crowdsourcing.

Crowdsourcing can also be broadly categorised as paid and unpaid crowd work based on the rewards received by workers. Paid work corresponds to crowdsourcing tasks where workers receive monetary rewards typically through a crowdsourcing platform that facilitates the payment process. Unpaid or voluntary crowd work is also completed in popular platforms and projects like Wikipedia,¹ Moral Machine [7], Crowd4U [83], Zooniverse,² and Test My Brain [50]. However, there are key distinctions in how you motivate unpaid and paid crowd work [55, 117, 140]. For example, in Test My Brain, workers get personalised feedback that helps them learn more about their mind and brain. In this review, we primarily focus on methods and literature that investigate paid crowdsourcing tasks on commercial crowdsourcing platforms.

When we consider the type of work available on crowdsourcing platforms, they can range from micro tasks [31] such as labelling, ranking, and classification to complex and long term tasks like software and web development tasks [151]. Our survey focuses on crowdsourcing techniques concerning tasks that can be completed in a single session, which constitutes the bulk of available crowd work.

3 QUALITY ENHANCEMENT IN CROWDSOURCING

As crowdsourcing typically relies on contributions from a diverse workforce where task requesters have limited information on the workers, it is important to employ data quality improvement measures [23]. In this section, we provide an overview of data quality in crowdsourcing.

In crowdsourcing, data quality is typically quantified via different attributes such as task accuracy, the response time of collected data, and cost-efficiency. Different quality improvement methods aim to improve one or more quality attributes. For example, the accuracy of a translation task can be enhanced in a cost-effective manner by employing workflow changes [5].

We note that quality improvement methods can differ from one another based on the following characteristics.

- *Applicability*: A quality improvement method can work for a specific type of task, a broader range of tasks or across all types of tasks. Universal methods are highly desired yet can be costly and difficult to implement. For example, certain question assignment methods [43, 84] only work for multi-class labelling tasks. In contrast, worker filtering based on approval rate works for most tasks when worker-history is available.
- *Complexity*: Some quality improvement methods involve complex implementations that require substantial time and effort. Such methods are not suitable for one-time jobs. For example, it is not straightforward to implement crowd workflows that facilitate real-time discussions among workers [19, 76].
- *Effectiveness*: The effectiveness of quality improvement methods also varies. The effectiveness of a method can be quantified by measuring the quality attributes.
- *Cost*: There is an inherent cost attached to each quality improvement method. It is explicit for some methods (e.g., issuing bonus payments to workers), while others have indirect costs (e.g., infrastructure cost to capture and analyse worker behaviour data).

Generally, task requesters prefer quality improvement methods that are low in complexity, highly effective, economical, and broadly applicable. However, methods that satisfy all these quality needs are scarce, and task requesters typically select quality improvement methods based

¹<https://www.wikipedia.org>.

²<https://www.zooniverse.org>.

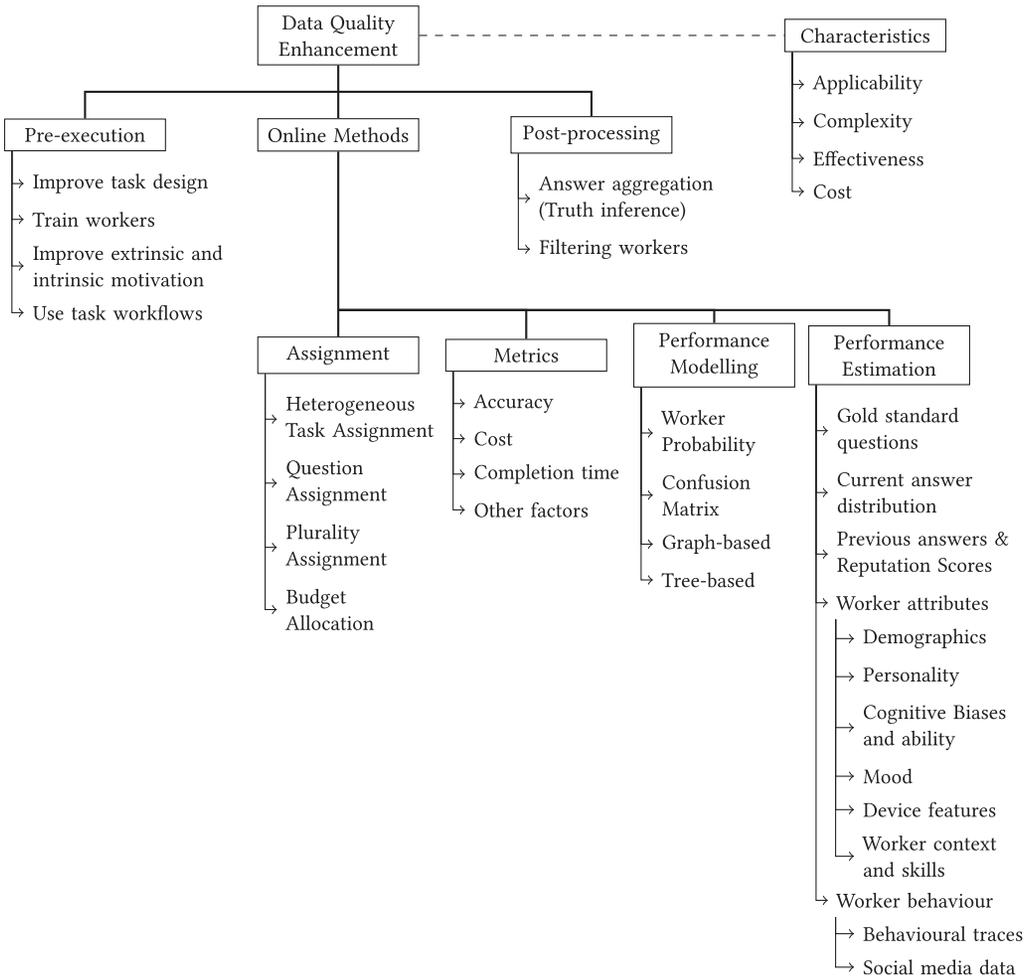


Fig. 1. Overview of quality enhancement methods and related concepts discussed in the survey.

on the specific task at hand, time and budget constraints, quality requirement, and platform compatibility.

While there is a wide array of such quality enhancement techniques, based on the method execution phase, they can be broadly categorised into pre-execution methods, online methods, and post-processing techniques as detailed in Figure 1. Given the standard crowdsourcing workflow, task requesters consider and employ pre-execution methods before task deployment. Fundamentally, through these methods, requesters specify how the task should be presented and executed in the crowdsourcing platform. Next, online methods alter the crowd task execution by dynamically deciding parameters such as the number of labels to collect, worker-task assignment, and task reward. Finally, post-processing methods examine how we can obtain better outcomes by processing the gathered crowd input. In this survey, we are primarily interested in online methods; however, we briefly summarise pre-execution and post-processing methods in the following sub-sections. In addition, Figure 1 provides an overview of different concepts and categories related to online assignment.

3.1 Pre-execution Methods

Data quality improvement methods employed at the pre-execution phase involve improving how workers interact with the task in terms of task design and crowdsourcing workflows.

3.1.1 Task Design and Crowdsourcing Workflows. Improving task design based on design guidelines and crowdsourcing best practices is one of the most well-known quality improvement methods. Research shows that clear task descriptions [49], data semantics or narratives that provide task context [35], and enhanced task user interfaces that improve usability [1, 4] and reduce cognitive load [3] elevate data quality.

The outcomes of methods relating to task design can vary depending on the task itself. For example, Find-Fix-Verify [11] is a workflow introduced for writing tasks such as proofreading, formatting, and shortening text. Iterate and vote is another design pattern where we ask multiple workers to work on the same task in a sequential manner. Little et al. [110] shows that the iterate and vote method works well on brainstorming and transcription tasks. Similarly, under map-reduce, a larger task can be broken down into discrete sub-tasks and processed by one or more workers. The final outcome is obtained by merging individual responses [21, 102].

Many other complex workflows have been proposed. For instance, the assess, justify, and reconsider [39] workflow improves task accuracy by 20% over the majority vote for annotation tasks. Several extensions to this method have been proposed, such as introducing multiple turns [19, 145]. Annotate and verify is another workflow that includes a verification step. Su et al. [152] show that data quality in a bounding box task is improved when they employ the annotate and verify method with two quality and coverage assessment tasks followed by the drawing task [152].

More complex workflows that facilitate real-time group coordination [19, 145] can be challenging to incorporate into a crowdsourcing platform. Other variants include tools that allow workers [105] and task requesters (e.g., Retool [18] and CrowdWeaver [100]) to design custom workflows. There is limited work that explores how to build and manage the crowdsourcing pipeline when employing a task workflow [154]. For example, the reward for each step can be dynamically adjusted to efficiently process the overall pipeline [122]. On the contrary, some work argues that static crowdsourcing workflows are limited in terms of supporting complex work and calls for open-ended workflow adaptation [138].

Other related task design and workflow improvements include gamification [54, 125] and adding breaks or micro-diversions [22].

3.1.2 Feedback and Training. Providing feedback to workers based on their work can improve the data quality in crowdsourcing. Dow et al. [37] report that external expert feedback and self-assessment encourages workers to revise their work. Dow et al. [37] highlight three key aspects of feedback for crowd work. “Timeliness” indicates when the worker gets feedback (i.e., synchronously or asynchronously). The level of detail in the feedback or “specificity” can vary from a simple label (e.g., approve, reject) to more complex template-based or detailed one-to-one feedback. Finally, there is “source” or the party giving feedback, which can be experts, peer workers, the requester, or the worker themselves.

In a peer-review setup, the process of reviewing others’ work has also been shown to help workers elevate their own data quality [170]. Similarly, workers achieve high output quality when they receive feedback from peers in an organised work group setting [160]. While expert and peer feedback are effective in improving data quality, it is challenging to ensure the timeliness of feedback, which is important when implementing a scalable feedback system.

It is also possible to deploy a feedback-driven dedicated training task and let workers complete multiple training questions until they achieve a specified data quality threshold. Park et al. [132]

report that such a mechanism can be effective in crowdsourcing tasks that involve complex tools and interfaces. However, training or feedback may also bias the task outcome depending on the specific examples selected for the training/feedback step [107]. Feedback can also be used to explain unclear task instructions. For example, prior work by Manam and Quinn [116] proposes a Q&A and Edit feature that workers can use to clarify and improve task instructions or questions.

Other similar work tools that can potentially help improve data quality include third-party web platforms, browser extensions, and scripts (e.g., Turkopticon [85] and Panda Crazy³) [93]. These tools provide additional information for workers to avoid substandard tasks and make their work more efficient.

3.2 Online Methods

While pre-execution methods focus on priming the task and workers, online methods aim to increase data quality by dynamically changing task deployment parameters and conditions like matching workers with compatible and relevant tasks. In this survey, we primarily focus on such online assignment methods, which we discuss in detail in Sections 4, 5, and 6.

3.3 Post-Processing Methods

Post-processing methods are employed after workers complete the entire batch of tasks in the crowdsourcing platform. A large portion of post-processing methods falls under answer aggregation techniques. We also discuss several other methods, including filtering workers.

3.3.1 Aggregating Answers. Typically in crowdsourcing, we obtain multiple answers for each question. Once all the answers are collected, we need to aggregate them to create the final answer for each question. This process is also known as truth inference in crowdsourcing. There are many ways to aggregate answers, and task requesters may opt for different strategies depending on the task and data quality needs.

Majority voting is the most simple and naive, yet widely used, approach for answer aggregation [167]. However, majority vote can fail when only a handful of highly accurate workers provide the correct answer. Prior work has proposed many extensions to majority voting. For example, instead of calculating the majority vote, the labels can be aggregated to a score that reflects the level of agreement [167]. Then, we can calculate the best threshold value to obtain the final answer. A training set or a gold-standard question set can be used when determining the threshold.

Zhuang et al. [171] examined the bias that can be introduced into crowdsourcing when a worker provides answers to multiple tasks grouped into a batch, which is a common mechanism employed to reduce cost and improve convenience for the worker. They proposed an alternative to majority voting, which could result in improved accuracy when batching is present. Ma et al. [115] proposed a truth inference method that is able to account for the varying expertise of workers across different topics.

For rating and filtering tasks, Das Sarma et al. [24] proposed an algorithm for finding the global optimal estimates of accurate task answers and worker quality for the underlying maximum likelihood problem. They claim their approach outperforms **Expectation Maximisation–(EM)** based algorithms when the worker pool is sufficiently large. Further, in an extensive survey on truth inference, Zheng et al. [167] evaluate the performance of different truth inference algorithms.

3.3.2 Clustering. Kairam and Heer [89] proposed an automated clustering-based method as a design pattern for analysing crowd task responses. Using entity annotations of Twitter posts and

³<https://github.com/JohnnyRS/PandaCrazy-Max>.

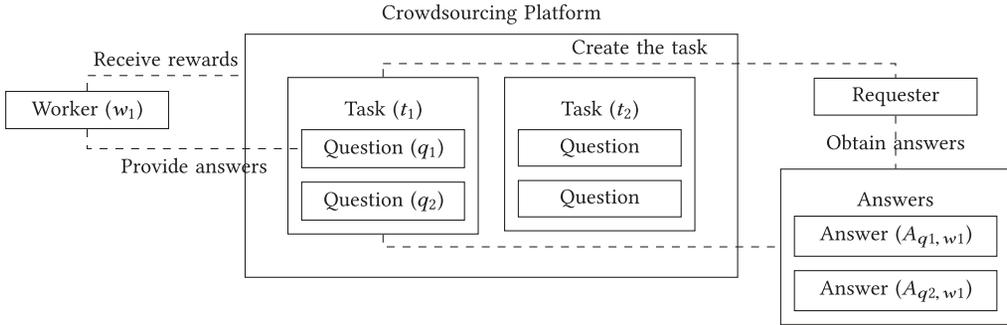


Fig. 2. Components of a standard crowdsourcing workflow and the relationships among them.

Wikipedia documents, they identified systematic areas of disagreement between groups of workers that can be used to identify themes and summarise the responses.

3.3.3 Filtering Answers. After data collection, we can also remove specific responses to improve the data quality. For example, if we are able to identify malicious workers who may submit purposely inaccurate or incomplete responses, then we can filter all the answers provided by such users during the aggregation process. Instead of using worker responses as the sole quality signal, Moshfeghi et al. [126] propose a method that uses task completion time to identify careless workers. Similarly, post hoc worker filtering is also possible after estimating worker accuracy through different techniques, such as analysing worker behavioural traces [65, 142] and the worker network [104]. In Section 5.3, we discuss estimation methods in detail. Furthermore, data quality can be impacted when workers use bots to provide automated responses or collude with other workers to share information [17, 32]. KhudaBukhsh et al. [99] propose an unsupervised collusion detection algorithm that can help identify such workers and remove corresponding responses. It is also possible to detect colluding workers by analysing contribution similarity [91]. In addition, sybils or bots can be identified by estimating worker similarity and clustering them into groups [163].

4 TASK ASSIGNMENT PROBLEMS

Before we examine online methods in detail, it is important to identify the different stakeholders and parameters involved. We explain the crowdsourcing workflow (Figure 2), involved entities, and different parameters that can be optimised in an online setting for task assignment purposes.

- **Requester:** A person who posts tasks on a crowdsourcing platform. Requesters reward the workers through the platform when they provide answers to their task.
- **Worker:** A person who completes tasks on a crowdsourcing platform in return for a reward. There is a large body of literature that examines characteristics of worker population [30, 141], work practices [162], and challenges faced by workers [144].
- **Task:** A collection of questions of the same task type. Prior work [48] has identified different task categories, such as verification and validation, interpretation and analysis, content creation, surveys, and content access.
- **Question:** An individual question within a task. For example, in an Audio Annotation task, this would be an audio clip that requires an annotation. An arbitrary number of answers can be collected for each question. Typically, this threshold or the number of labels required for each question is pre-determined by the requester.

Table 2. Notations Used in the Survey

Set of workers	$W = \{w_1, \dots, w_n\}$
Set of tasks	$T = \{t_1, \dots, t_n\}$
Set of questions for task t	$Q_t = \{q_1, \dots, q_n\}$
A task assignment	$\{t, w\}$
A question assignment of question q and worker w	$QA_{q,w}$
An answer provided by worker w to question q	$A_{q,w}$
Reward or payment for a question q	R_q

- *Answer*: The answer provided by a specific worker to a specific question. Answer could take different forms depending on the task (e.g., a label “Positive” in sentiment analysis). Typically in crowdsourcing, multiple workers provide answers for the same question. Numerous metrics such as accuracy and response time can be used to measure the quality of an answer.
- *Reward*: There can be intrinsic and extrinsic rewards [140]. The main reward mechanism used in crowdsourcing includes providing a pre-specified base payment and bonus payments issued at requesters discretion.
- *Crowdsourcing Platform*: Interaction between workers and task requesters is often managed by a third-party platform. For example, Amazon Mechanical Turk, Appen, Prolific, and Toloka are commercial crowdsourcing platforms that charge a fee from task requesters for managing the crowdsourcing workflow.

As detailed in Table 2, we use a consistent notation throughout the survey to describe different assignment problems.

While the interaction between entities detailed above can vary depending on the specific crowdsourcing platform, next we summarise a typical crowdsourcing workflow. Task requesters first post their tasks in a crowdsourcing platform, with specific instructions and rewards for successful completion. Workers who have already signed up in the platform can browse and start working on tasks that they are eligible for. Eligibility constraints (e.g., location, skill and quality requirements) are often set by requesters or the crowdsourcing platform itself. Finally, when the work is completed, requesters can obtain the worker input or data contributions from the platform and compute the final output. Optionally, they may indicate whether individual worker answers meet their expectation. For instance, requesters can “approve” or “reject” answers. The crowdsourcing platform then transfers the reward to workers. This is similar to a first-come-first-serve or a market model. Online assignment methods in crowdsourcing aim to alter this market model by directing workers to relevant and compatible tasks to increase the overall data quality. At a high level, we identify and examine four key assignment challenges; heterogeneous task assignment, question assignment, plurality assignment problem, and budget allocation.

4.1 Heterogeneous Task Assignment Problem

The aim of heterogeneous task assignment, or simply “task assignment,” is to select the best-suited task for a worker when there are different tasks available (e.g., Sentiment Analysis, Entity Resolution, and Classification).

Definition. Assume that we have a set of tasks $T = \{t_1, \dots, t_k\}$ and a set of workers $W = \{w_1, \dots, w_m\}$ where $|T| = k$ and $|W| = m$. Each task t may contain an arbitrary number of questions. To maximise the overall quality of the data we gather, for each worker $w \in W$, we aim to assign the task t' where the worker is more likely to produce results of better quality.

4.2 Question Assignment Problem

Select a specific number of questions from a task for a worker. For example, in a Twitter Sentiment Analysis task with 1,000 tweets, the aim is to find specific tweets to assign for each worker.

Definition. Assume that we have a set of questions $Q = \{q_1, \dots, q_k\}$ for a specific task t and a set of workers $W = \{w_1, \dots, w_m\}$ where $|Q| = k$ and $|W| = m$. To maximise the overall quality of the data we gather, for each worker, we aim to assign one or several questions where the worker is more likely to produce results of better quality.

4.3 Plurality Assignment Problem

Deciding on the optimal number of workers that should be assigned to each sub-task or question is known as plurality assignment problem. Typically in crowdsourcing platforms, requesters manually configure a fixed number as the number of workers to be assigned for each task.

Definition. Assume that we have a set of questions $Q = \{q_1, \dots, q_k\}$ for a specific task t and a set of workers $W = \{w_1, \dots, w_m\}$ where $|Q| = k$ and $|W| = m$. For each question $q \in Q$, multiple workers can provide answers (e.g., $A_{q,w_1}, A_{q,w_2}, \dots, A_{q,w_x}$). We want to determine the ideal number of answers needed for each question q .

4.4 Budget Allocation

The wide popularity of crowdsourcing is largely due to its economical nature when compared to other ways to acquiring large volumes of data. Hence, in addition to the data quality, budget allocation is an important factor in crowd work. Certain work considers budget allocation as part of the task or question assignment problem. For example, Assadi et al. [6] investigate task assignment with the aim of maximising the number of tasks allocated with a fixed budget.

5 WORKER PERFORMANCE ESTIMATION

Worker performance estimation is a critical step in online assignment process. If performance estimations are unreliable, then subsequent task, question, or budget assignment decisions will not lead to desired quality enhancements. In this section, we discuss different metrics that can be used for estimation, data structures utilised for worker performance modelling and ways of estimating the performance.

5.1 Performance Metrics

5.1.1 Accuracy. Task accuracy is the most widely used performance metric in crowdsourcing. Accuracy is typically a number between 0 (incorrect) and 1 (correct) and can be defined in different ways depending on the task. For instance, for a classification task with single correct answer, accuracy of each question would be 1 if the worker provides the correct label and 0 otherwise. In contrast, a distant metric can define the similarity between text for translation tasks, which results in a fraction. Other metrics that represent task accuracy include F-score [168], information gain [109] for multiple-choice tasks, mean Intersection over Union for image annotation tasks [131], and so on.

5.1.2 Cost. While there are different crowd pricing mechanisms discussed in the literature [150], in a typical crowdsourcing platform, there is a pre-specified cost attached to each collected answer. However, other costs such as bonus payments and platform fees (e.g., MTurk⁴) can

⁴<https://www.mturk.com/pricing>.

increase the total cost. Since crowdsourcing is often used for tasks with a large number of questions, cost is considered an important performance metric.

5.1.3 Task Completion Time. When we consider task completion, there are two key metrics: time that workers spend on completing each question (i.e., work time) and total time needed to complete a task job that contains a set of questions (i.e., batch completion time). Both metrics can be optimised in different ways. Minimising work time is particularly helpful for tasks that require workers with specific skills or backgrounds [119]. In addition to task assignment, task scheduling strategies also aim to optimise batch completion time [29]. Crowdsourcing platforms typically provide task time information to requesters and they can also set a maximum time limit for each question.

5.1.4 Other Factors. Another indirect performance metric is worker satisfaction. Prior work highlights a relationship between crowd worker satisfaction and turnover [13], which may have an impact on data quality in the long run.

Some task assignment methods also consider special properties depending on the task. For instance, privacy preservation is an important performance metric for audio transcription tasks [15]. Others have considered the fairness [51], worker survival, or likelihood to continue on tasks [103] and diversity in terms of worker properties [9].

5.2 Worker Performance Modelling

Based on the complexity and requirements of worker performance estimation method and the task or question assignment method, the literature proposes different ways to represent the quality of each worker, which we summarise below.

5.2.1 Worker Probability. The quality of each worker is modelled by a single attribute that describes the probability of the worker providing the true answer for any given question. This is a simple and widely adopted method [63, 113]. However, a single probability score is often insufficient to model the quality of the worker due to variations in question difficulty. The basic worker probability model can be extended by including a confidence value along with the probability value [86].

Instead of using a single probability value for all the tasks, worker probability can be modelled for each task (e.g., Reference [123]) or question within the task (e.g., Reference [43]). For example, quality of a specific worker could be 0.5 for sentiment analysis task and 0.8 for classification task.

5.2.2 Confusion Matrix. Confusion matrix is extensively used to model worker performance for multiple-choice questions where each question has a fixed number of possible answers (e.g., References [137, 158, 159]). Each cell (i, j) within the matrix indicates the probability of the worker answering the question with a label i given the true answer of the question is j . For the initialisation, each worker could be assumed a perfect worker, values could be drawn from a prior distribution, or values can be estimated using gold-standard questions.

5.2.3 Graph-based. In a graph-based model, workers or tasks are modelled as nodes in a graph (e.g., References [14, 139, 165]). Edges represent possible relationships among them. Different approaches are also possible. For instance, task assignments can be modelled as edges in a bipartite graph with both workers and questions as nodes (e.g., References [94, 112]).

5.2.4 Tree-based. A tree-based model is a slight variant of the graph-based model. For instance, Mavridis et al. [119] uses a skill taxonomy modelled as a tree where nodes represent elementary skills. Each worker also has a set of skills that they possess. A skill distance metric between the

Table 3. An Overview of Worker Performance Estimation Methods Used in Online Assignment Methods

Method	Literature	
Gold Standard Questions & Qualification Tests	[84], [113]	
Current Answer Distribution	[168], [98], [8], [136]	
Previous Answers & Reputation Scores	[133], [146]	
Worker Attributes	Demographics	[96], [148], [41], [30]
	Personality Tests	[95], [96], [114]
	Skills	[119], [106]
	Cognitive Tests	[52], [71], [72]
	Work Device Features	[44], [70]
	Worker Context	[82], [73]
Worker Behaviour	Behavioural Traces	[142], [65], [45], [60]
	Social Media Data	[33], [165]

required skills for the task and the given skills of a worker is considered as the worker quality value for the particular task.

5.3 Performance Estimation Methods

Before assigning tasks or questions to workers, we need to estimate the performance of each worker. Estimations can be obtained by using objective measures such as gold-standard questions, past/current task performance data, and qualification tests or by using worker characteristics or behavioural traits that are known to correlate with task performance. Table 3 organises prior work based on the performance estimation method.

5.3.1 Gold Standard Questions. Gold Standard Questions are questions with a known answer. It is common practice to use gold-standard questions to estimate worker performance [113]. Typically, gold questions are injected into the task to appear among regular questions such that workers are unable to anticipate or detect gold questions.

When implementing gold standards, it is essential to know how we can inject these questions systematically. Prior work by Liu et al. [111] investigates the optimum number of gold questions to use in a task. It is not beneficial to use a small number of gold-standard questions in a large question batch. Workers could then collectively identify and pay more attention to gold questions, making them ineffective as quality checks [16, 17]. Furthermore, creating ground-truth data is not straightforward, and crowdsourced tasks often do not have ground-truth data. Therefore, scalable and inexpensive methods of creating good gold data are necessary when using gold standards as a quality improvement method. Oleson et al. [129] present a programmatic approach to generate gold-standard data. They report that a programmatic gold method can increase the gold per question ratio, allowing for high-quality data without extended costs.

Instead of populating gold questions before the label collection, we can also validate selected answers using domain experts. For instance, Hung et al. [81] propose a probabilistic model for classification tasks that help us find a subset of answers to validate through experts. The method considers the output accuracy and detection of inaccurate workers to find the most beneficial answer to validate. In addition, we can use domain experts to generate reliable and high-quality gold data [67]. Finally, in addition to measuring worker performance, gold-standard questions can function as training questions that provide feedback to workers [46, 107].

5.3.2 Qualification Tests. Qualification tests contain a set of questions that workers need to complete before accessing the task. A qualification test can contain questions related to worker

experience, background, or skills that are needed for the actual crowdsourcing task [121]. For instance, a simple language skill test could be an appropriate qualification test for a translation task. A set of gold-standard questions can also be presented as a qualification task. As answers are known *a priori*, requesters can measure the performance in qualification test and allow a subset of workers to attempt the regular task. Crowdsourcing platforms such as MTurk supports qualification tests.

When using gold-standard questions as a qualification test, there should be sufficient coverage of the different questions included in a task. Similarly, the qualification test should be challenging, such that workers are unable to pass without understanding the task instructions fully.

When employing qualification tests, we can also ask workers to assess their own responses when ground-truth data are not available or automated assessment is not feasible. Gadiraju et al. [47] show that self-assessment can be a useful performance indicator when we account for varying levels of accuracy in worker self-assessments.

5.3.3 Using Current Answer Distribution. In an ongoing task, we can also use the current answer distribution to estimate worker accuracy. Expectation Maximisation [25] is one of the most commonly used estimation methods to gauge worker performance for multiple class labelling questions (i.e., multiple choice questions) [168]. The method examines all the current answers and iteratively updates worker quality values and task answers until they converge. Khan and Garcia-Molina [98] used a different approach that uses Marginal Likelihood Estimation. They report that compared to EM, Marginal Likelihood Estimation significantly reduces root mean squared error in predicting worker accuracy when there are few votes per worker. Raykar and Yu [136] considers a discrete optimisation problem and propose a Bayesian approach that can estimate a binary state that decides whether a worker is a spammer or not.

Estimating worker accuracy from current answer distribution is not exclusive to labelling tasks. Baba and Kashima [8] introduced a two-stage workflow with a creation and a review stage for tasks with unstructured responses, such as content generation and language translation. Their method uses the maximum a posteriori inference to estimate the accuracy and model parameters.

5.3.4 Previous Answers and Reputation Scores. Once crowdsourcing tasks are completed, task requesters can indicate whether they are satisfied with worker responses. Similarly, we can also test if worker responses agree with the majority response. Such signals can be incorporated into a reputation score (e.g., approval rate in Amazon Mechanical Turk platform) and can be used to estimate the worker performance [133]. In addition, research shows that trust relationships among workers can be leveraged as reputation scores [146].

5.3.5 Worker Attributes. When looking at task or question assignment from the workers' perspective, several worker attributes have been shown to have an impact on crowd task performance.

- **Demographics:** While there is no evidence to support a direct link between demographics and worker performance, literature reports on specific tasks where demographics may influence the performance [96, 148]. Importantly, researchers note that local knowledge [53, 55], language [156], and work tools [44] of crowd workers, and the differences in pay rates [10, 77] can lead to location-based performance variations. For example, in a content analysis task that involves assessing US political blogs, Shaw et al. [148] have shown that US workers unsurprisingly perform significantly better than Indian workers. In contrast, in an attempt to examine the preference for games over conventional tasks in relevance labelling, Eickhoff et al. [41] reported no significant difference in the performance of workers from the US and India in Amazon Mechanical Turk. Other demographic factors such as age [96] can also influence performance in specific tasks. Other work have also shown that worker demographics can introduce biases to the data collected [30, 69].

- *Personality*: Kazai et al. [95] analysed crowd users based on five personality dimensions introduced by Goldberg [87] known as the “Big Five.” They further segmented workers into five types (Spammer, Sloppy, Incompetent, Competent, and Diligent) based on the personality and reported a significant correlation between the worker type and the mean accuracy of the worker. In a subsequent study, Kazai et al. [96] also reported that the Big Five personality traits—openness and conscientiousness—are correlated with higher task accuracy. Lykourantzou et al. [114] also examined the effect of personality on the performance of collaborative crowd work on creative tasks. They created 14 five-person teams: balanced (uniform personality coverage) and imbalanced (excessive leader-type personalities), considering only the outcome of “DISC” [118] (dominance, inducement, submission, compliance) personality test and reported that balanced teams produce better work in terms of the quality of outcome compared to imbalance teams.
- *Cognitive Biases*: The study by Eickhoff [40] investigates cognitive biases and shows that cognitive biases negatively impact crowd task performance in relevance labelling. Cognitive biases are known as systematic errors in thinking and can impact peoples everyday judgements and decisions.
- *Cognitive Ability*: Alagarai Sampath et al. [3] experiment with task presentation designs relating to cognitive features such as visual saliency of the target fields and working memory requirements. The study conducted on MTurk uses a transcription task and reports design parameters that can improve task performance. Goncalves et al. [52] investigated the impact of the cognitive ability of crowd worker performance and demonstrated that performance can be predicted from the results of cognitive ability tests. In their study, they used 8 cognitive tests that included visual and fluency tasks and 8 different crowdsourcing task categories attempted by 24 participants in a lab setting. However, they used time-consuming and paper-based cognitive tests from ETS cognitive kit [42] that are not practical for an online setting. Hettiachchi et al. [71] investigate the effect of cognitive abilities on crowdsourcing task performance in an online setting. The work leverages the three executive functions of the brain (inhibition control, cognitive flexibility and working memory) [27] to describe and model the relationship between cognitive tests, and crowdsourcing tasks. A subsequent study [72] proposes a dynamic task assignment approach that uses cognitive tests.
- *Mood*: Prior work has also investigated if workers’ mood has any impact on the crowdsourcing task performance [172]. While there is no evidence that shows a direct link between mood and task accuracy, the study reports that workers in a pleasant mood exhibit higher perceived benefits from completing tasks when compared to workers in an unpleasant mood.
- *Work Device Features*: Gadiraju et al. [44] show that crowd work device and its characteristics such as screen size and device speed have an impact on data quality. The research also highlights that the negative impact of bad user interfaces is exacerbated when workers use less suitable work devices. In addition, device sensing capabilities and battery level can also impact the quality of crowd contributions [68]. Hettiachchi et al. [70] explore voice-based crowdsourcing, where workers complete crowd tasks through smart speakers and investigate if there is a performance difference compared to regular crowd work through desktop computers.
- *Worker Context*: Other contextual factors concerning the worker’s current situation can also impact crowd task performance. Ikeda and Hoashi [82] show that task completion rate decreases when workers are busy or with other people. Also, worker context is a critical performance estimator for task assignment in spatial crowdsourcing, where tasks relate to a specific location [61]. Hettiachchi et al. [73] investigate workers’ willingness to

accept crowd tasks to understand the impact of context when tasks are available through a multitude of work devices.

- *Skills*: Prior work by Mavridis et al. [119] estimates worker performance using a distance measure between the skills of the worker and the skills required for the specific task. They use a taxonomy-based skill model. Similarly, Kumai et al. [106] model each skill with a numeric value. For instance, 1 minus the average word error rate of a worker's typing results can represent their typing skill.

5.3.6 Worker Behaviour. Prior work shows that worker behaviour data can be used to estimate worker performance [45, 60, 65, 142]. Rzeszotarski and Kittur [142] proposed “task fingerprinting,” a method that builds predictive models of task performance based on user behavioural traces. Their method analyses an array of actions (e.g., scrolling, mouse movements, key-strokes) captured while the user is completing crowdsourcing tasks. Task fingerprinting has been shown to be effective for image tagging, part-of-speech classification, and passage comprehension tasks in Amazon Mechanical Turk.

Han et al. [65] also reported that most of the worker behavioural factors are correlated with the output quality in an annotation task. Their method includes several additional features compared to the task fingerprinting method [142] and uses four types of behavioural features: temporal, page navigation, contextual, and compound. In a different approach, Kazai and Zitouni [97] show how we can use the behaviours of trained professional workers as gold-standard behaviour data to identify workers with poor performance in relevance labelling.

While other methods [65, 142] aim to classify workers into either “good” or “bad” categories, Gadiraju et al. [45] classify workers into five categories using behavioural traces from completed HITs. The study shows that significant accuracy improvements can be achieved in image transcription and information finding tasks by selecting workers to tasks based on given categories. To predict task and worker accuracy in relevance labelling tasks, Goyal et al. [60] uses action-based (e.g., mouse movement in pixels in horizontal direction, total pixel scroll in vertical direction) and time-based (e.g., fraction of the total time that was spent completing the HIT, mean time between two successive logged click events) features in their predictive model. Goyal et al. [60] argue that worker behaviour signals captured in a single session can be used to estimate the work quality when prior work history is unavailable.

Behavioural data like social media interests captured outside the crowdsourcing platforms have also been used to predict task performance [33]. While this can be an interesting direction that attempts to create a global profile of the crowd worker, current strict privacy regulations would make practical implementation almost impossible.

5.3.7 Using a Combination of Estimators. Rather than using a single performance estimator, it is also possible to use a combination of different estimators. For instance, most of the EM-based methods use gold-standard questions for initial estimation. Similarly, Barbosa and Chen [9] introduces a framework where the worker pool for each task can be constrained using multiple factors such as demographics, experience, and skills. Their results show that worker selection with appropriate uniform or skewed populations helps mitigate biases in collected data.

5.4 Challenges and Limitations in Performance Estimation

While prior work reports promising results on using various worker performance estimation methods, there are many limitations when we consider implementation and broader adoption of such method.

Perhaps the most well-known estimation method is the use of gold-standard questions. However, there are several fundamental limitations. First, gold-standard questions are not broadly

available for all tasks (e.g., tasks with subjective responses). Second, it can be costly to generate good gold questions. Third, gold questions are also susceptible to adversarial attacks. In an attack, workers detect and mark gold-standard questions through various third-party tools such that subsequent workers can pay more attention to gold-standard questions to amplify their measured performance [17]. Despite such limitations, the use of gold-standard questions is an effective quality control method applicable to a broader range of tasks.

Worker attributes are also widely used to estimate the worker performance. Attributes like cognitive ability, personality, and skills are preferred, as they can be extended to estimate task performance across a wider range of tasks. Similarly, task requesters often use demographics (e.g., location, age, education level) as it is straightforward to use them. However, there are notable challenges in integrating certain worker attributes into a task assignment system. For example, attributes like demographics are self-reported by workers, allowing workers to provide incorrect information to gain undue advantages. Comprehensive personality tests are time-consuming, and there is also the possibility for workers to manipulate the outcome. Similarly, less competent crowd workers tend to overestimate their performance in self-assessments [47].

In addition, demographics-based performance estimation could lead to biased or unfair assignment and discrimination by task requesters leading to fewer tasks of a particular type assigned to workers with specific demographic attributes (e.g., gender, ethnicity, race) [66]. Such unethical approaches and problems should be addressed by crowdsourcing platforms, as well as by researchers.

Numerous complications exist when concerning the use of worker skills [106, 119]. Workers need to list down their skills and such information should be available at platform level. We have to either assume that worker input related to skills are accurate or validate such information. Skill assessment can be a lengthy process increasing the barrier of entry for new workers. Also, requesters have to define which skills are required when creating new tasks.

While worker activity tracking [45, 60, 65, 142] has shown promising results, there are several practical limitations. First, such implementations often run as browser-scripts and can make the crowdsourcing platform interface resource intensive. This in turn can limit the accessibility of crowdsourcing platforms, particularly for workers with computing devices with limited capacities and low bandwidth internet connectivity. Second, behavioural data collection, data storage, and performance estimation can be computationally intensive for the back-end infrastructure of the crowdsourcing platforms, thus incurring additional costs. Third, there are privacy concerns with regard to tracking and storing activity data.

6 TASK ASSIGNMENT METHODS

In this section, we discuss methods or frameworks that actively prevent contributions of sub-par quality by implementing various quality control mechanisms. In contrast to post-processing techniques, task assignment or routing methods can significantly reduce the overall number of answers required to obtain high-quality output for crowd tasks. Thus, they can bring a financial benefit to task requesters. Also, task assignment can increase the compatibility between worker capabilities and task needs, potentially leading to increased worker satisfaction.

Literature presents a number of task assignment algorithms or frameworks that can be integrated with, or used in place of existing crowdsourcing platforms. They consider different quality metrics (e.g., accuracy, task completion time) and implement one or more quality improvement techniques (e.g., gold-standard questions [38], removing or blocking erroneous workers [98]) to enhance the metrics. The primary motivation behind each assignment method can also be divergent. For example, some methods aim to maximise the quality of the output (e.g., References [43, 143, 168]) while other methods attempt to reduce the cost by achieving a reasonable accuracy with a minimum number of workers (e.g., Reference [98]).

We organise prior work under task assignment, question assignment and plurality problems we outlined in Section 4. Table 4 provides a brief summary of the worker performance estimation and assignment strategy of each method we discuss in this section.

6.1 Heterogeneous Task Assignment

As crowdsourcing platforms contain a variety of tasks (e.g., sentiment analysis, classification, transcription), heterogeneous task assignment focuses on matching different task types with workers. Heterogeneous task assignment can be particularly useful in cases where “expert” workers must be allocated for more difficult tasks [75]. In addition to heterogeneous task assignment, the crowdsourcing literature also explores question assignment, where questions within the same task (e.g., different questions of sentiment analysis task) are assigned to different workers to maximise the performance gain. We also review question assignment methods in Section 6.2.

Task assignment involves multiple steps. First, worker performance is modelled and estimated using different methods discussed in Section 5. Then, the task assignment process is carried to maximise the potential gain in terms of a specific performance criteria. For instance, one task assignment method could achieve modest data quality gains while minimising the overall cost. In contrast, another method could aim to achieve the highest possible data quality with a set budget.

Ho and Vaughan [75] propose a task assignment method based on the online primal-dual framework, which has been previously utilised for different online optimisation problems. The proposed Dual Task Assigner algorithm assumes that workers with unknown skills request tasks one at a time. In the study, researchers use three types of ellipse classification tasks to account for different expertise levels and use a translation task to simulate different skills. However, their approach assumes that the requester can immediately evaluate the quality of completed work. This vastly limits the applicability of their approach in a real-world crowdsourcing problem. Ho et al. [74] further investigate heterogeneous task assignment in classification tasks with binary labels. However, for the assignment, they use gold-standard questions of each task type to estimate the accuracy of the workers.

We can also examine task assignment from the requester perspective. Assadi et al. [6] propose an online algorithm that can be used by a requester to maximise the number of tasks allocated with a fixed budget. In a different approach for task assignment, Mo et al. [123] apply a hierarchical Bayesian transfer learning model. They use the historical performance of workers in a similar or different type of tasks to estimate the accuracy of the new tasks. Their experiment with a real-world dataset shows the effectiveness of the proposed approach when transferring knowledge from related but different crowd tasks (e.g., questions on sports vs makeup and cooking). However, their real-world evaluation is limited to a single scenario with one source task and one target task.

While most methods focus on a predefined set of tasks, Dickerson et al. [28] examine task assignment when tasks are not known *a priori*. Their work proposes a novel theoretical model, called Online Task Assignment with Two-Sided Arrival, where both workers and tasks arrive in an online manner.

Data collected outside crowdsourcing platforms can also be used to match tasks with workers. Difallah et al. [33] present a system where tasks are allocated based on worker profile data such as interested topics captured from a social media network. Similarly, Zhao et al. [165] propose “Social Transfer graph” for task matching. They demonstrate how tasks on Quora can be matched with Quora users’ by extracting respective users’ Twitter profile data (i.e., tweets and connections). The general applicability of such methods raises numerous practical and ethical considerations.

Mavridis et al. [119] introduced a skill-based task assignment model. Worker performance is estimated using a distance measure between the skills of the worker and the skills required for the

Table 4. An Overview of Worker Performance Estimation and Assignment Strategies of Assignment Methods

Task Question Plurality	Assignment Problem	Reference	Performance Estimation	Assignment Strategy	Method Maturity and Evaluation
✓		[75]	Requesters manually evaluate the answers.	Based on the online primal-dual framework.	Basic Research. Offline evaluation using real-world crowdsourcing data.
✓		[74]	Using gold-standard questions.	By extending online primal-dual methods.	Basic Research. Offline evaluation using simulations and synthetic data.
✓		[6]	Using bids provided by workers.	Maximises the number of tasks allocated within a budget.	Basic Research. Offline evaluation using simulations and synthetic data.
✓		[123]	Estimate using the performance in other tasks.	Through a hierarchical Bayesian transfer learning model.	Basic Research. Offline evaluation using synthetic and real-world crowdsourcing data.
✓		[28]	Use historic records to learn quality distributions.	Model workers and tasks in a bipartite graph and use an adaptive, non-adaptive or greedy method to assign tasks.	Basic Research. Offline evaluation using real-world crowdsourcing data.
✓		[72]	Estimated using cognitive test outcomes.	Select workers to maximise gain in accuracy.	Prototype Implementation. Online dynamic evaluation with crowd workers.
✓		[33]	Using interested topics captured from social media.	Rank available workers through category-based, expert-profiling and semantic-based assignment models.	Prototype Implementation. Online dynamic evaluation with crowd workers.
✓		[119]	Through a distance measure between worker skills and the skills required for tasks.	Targets skill compatibility. Assigns specialised tasks to workers with fewer skills first.	Basic Research. Offline evaluation using synthetic and real-world crowdsourcing data.
✓		[34]	Assumes that context-switching reduces worker satisfaction and performance.	Scheduling tasks to maximise the likelihood of a worker receiving a task that they have recently worked on.	Prototype Implementation. Online dynamic evaluation with crowd workers.
✓		[29]	Assumes that context-switching reduces worker satisfaction and performance.	Schedule tasks prioritising currently running jobs and workers getting familiar work.	Prototype Implementation. Online dynamic evaluation with crowd workers.
✓		[15]	Estimate the loss of private information	A graph-based method that maintains privacy without starving the on-demand workforce.	Basic Research. Offline evaluation using synthetic and real-world data.
✓		[106]	Estimate worker skills using a qualification task.	Form groups of workers based on skill balance and worker re-assignments.	Prototype Implementation. Online evaluation with crowd workers.
✓		[83]	Worker specified task interest and other factors such as skills.	Uses different strategies depending on the task collaboration scheme.	Prototype Implementation. No evaluation.
✓		[147]	Assumes that expertise of each worker is a known numerical value.	Sequential assignment based on budget, data quality and latency needs.	Prototype Implementation. Offline evaluation using synthetic and real-world data, and limited online evaluation with crowd workers.

(Continued)

Table 4. Continued

Task	Assignment Problem		Reference	Performance Estimation	Assignment Strategy	Method Maturity and Evaluation
	Question	Plurality				
✓	✓	[113] CDAS	Injecting gold-standard questions.	Estimate the required answer count and use early termination.	Prototype Implementation. Online dynamic evaluation with crowd workers.	
✓		[98] CrowdDQS	Marginal likelihood curve estimation.	Maximise gain in accuracy.	Prototype Implementation. Online dynamic evaluation with crowd workers.	
✓		[43] iCrowd	Static gold-standard questions & task similarity.	Save questions for most accurate workers.	Prototype Implementation. Online dynamic evaluation with crowd workers.	
✓		[143] OSQC	Hybrid gold plurality algorithm	Multi-rule quality control.	Basic Research. Offline evaluation using real-world crowdsourcing data.	
✓		[20] OKG	Statistical inference with Beta distribution priors.	Maximise gain in accuracy.	Basic Research. Offline evaluation using synthetic and real-world crowdsourcing data.	
✓		[168] QASCA	EM.	Maximise gain in accuracy or F-score.	Prototype Implementation. Online dynamic evaluation with crowd workers.	
✓		[84] Quizz	Estimate using only gold-standard question responses.	Maximise information entropy.	Prototype Implementation. Online dynamic evaluation with crowd workers.	
✓		[51]	Estimate using limited gold-standard questions.	Maximise gain in accuracy while satisfying budget, fairness and diversity constraints.	Basic Research. Offline evaluation using real-world crowdsourcing data.	
✓		[124]	Using gold-standard questions.	Estimate plurality form a greedy algorithm that assumes that answer quality increases monotonically at a decreasing rate with its plurality.	Basic Research. Offline evaluation using synthetic and real-world crowdsourcing data.	
✓		[149]	By modelling task difficulty and worker skills.	Through an incremental Bayesian model that re-evaluate answer quality at each stage.	Basic Research. Offline evaluation using real-world crowdsourcing data.	
✓		[155]	By iteratively estimating worker expertise and question difficulty.	Batch assignment maximising the number of questions completed in each batch.	Basic Research. Offline evaluation using synthetic and real-world crowdsourcing data.	
✓		[2]	Assumes the past performance of a worker is known.	Decide on when to stop assigning another worker.	Basic Research. Offline evaluation using synthetic and real-world crowdsourcing data.	

specific tasks. The method attempts to assign the most specialised task first to the workers with the lowest number of skills based on the distance measure.

Task assignment can be challenging for more complex and collaborative tasks. Ikeda et al. [83] propose a task assignment framework that can decompose complex tasks and support sequential, simultaneous and hybrid worker collaboration schemes. Their assignment strategy selects a worker based on interests indicated by workers and their eligibility calculated using the project description and worker human factors (e.g., language skills). In contrast, Schmitz and Lykourantzou [147] look at non-decomposable macro-tasks like document drafting. They propose a sequential assignment model, where multiple workers attempt a task on a fixed time-slot, one

after the other. At the end of each iteration, the next worker is selected if the task does not meet the desired quality threshold.

Instead of assigning tasks on the fly, it is also possible to schedule them when tasks are known *a priori*. Prior work by Difallah et al. [34] investigates task scheduling in crowdsourcing platforms and shows that scheduling can help minimise the overall task latency, while significantly improving the worker productivity captured through average task execution time. Research also highlights that scheduling is useful in ensuring tasks are fairly distributed across workers [29].

Addressing the growing concerns on crowdsourcing sensitive tasks like transcribing audio scripts, Celis et al. [15] examined task assignment with regard to tradeoff in privacy. To preserve content privacy, we need to ensure that not too many parts of the same job are assigned to the same worker. They introduced three settings: PUSH, PULL, and a new setting, Tug Of War, which aims to balance the benefit for both workers (by ensuring they can attempt a reasonable number of questions) and requesters (by minimising the privacy loss).

Instead of assigning tasks to individual workers, Kumai et al. [106] investigate the worker group assignment problem, where task requesters should select a group of workers for each task. They represent the worker accuracy using skills estimated through a qualification task and then forms groups based on three strategies that consider the skill balance among groups and the number of worker re-assignments.

6.2 Question Assignment

The aim of question assignment is to match workers with questions within a task such that we can obtain high-quality output. Unlike in heterogeneous task assignment, we need to estimate worker performance and allocate tasks as workers complete submit answers to individual or batches of questions. Zheng et al. [168] present a formal definition of question assignment problem in crowdsourcing and show that optimal question assignment is an NP-hard problem.

Question assignment involves several fundamental steps. First, we obtain a set of questions that are available to be assigned. Such candidate questions should not have been previously assigned to the current worker and should have available assignments with respect to the maximum number of answers required. Second, we estimate the performance gain (in terms of accuracy, for example) for each candidate question. Third, we select a subset of questions to be assigned to the given workers.

Baseline approaches for question assignment are random assignment or a round-robin assignment. Typical crowdsourcing platforms use these baseline approaches for question assignment.

6.2.1 Assigning Questions to Workers in a Sequential Manner. The question assignment problem can vary depending on the worker arrival assumption. The most practical problem is how to find a suitable question or a specific number of questions for an individual worker given a set of candidate questions. A naive way to assign questions is to enumerate all feasible assignments, calculate the performance gain for each assignment, and then choose the assignment with the maximum performance gain. However, this method is computationally expensive and is not practical for typical crowdsourcing platforms where each task has a large number of questions.

Zheng et al. [168] proposed a question assignment framework (QASCA) that attempt to maximise either accuracy or F-score. For assigning k questions based on accuracy, the paper proposes the Top-K benefit algorithm, which calculates the gain in expected number of correct answers for each question in candidate set and pick the questions that have the highest benefits. The algorithm has a time complexity of $O(n)$ where n is the number of questions in the candidate set. A more complex online algorithm is presented for assigning questions based on F-score.

“CrowdDQS,” proposed by Khan and Garcia-Molina [98], is a dynamic question assignment mechanism that examines most recent votes and selectively assigns gold-standard questions to workers to identify and removes workers with poor performance in real time. They claim the proposed system that integrates seamlessly with Mechanical Turk can drastically reduce (up to 6 times) the number of votes required to accurately answer questions when compared to a round-robin assignment with majority voting. The proposed question assignment method aims to maximise the potential gain. The algorithm greedily chooses a question from the candidate set whose confidence score stands to increase the most if another answer is obtained from the considered worker.

Another dynamic question assignment method proposed by Kobren et al. [103] uses the worker survival metric (a user’s likelihood of continuing to work on a task). Survival score is formulated using different measures such as accuracy, response time, and the difficulty of recently completed questions. The framework assigns questions to workers to achieve higher worker engagement and higher value for the task requester. Modelled using the Markov decision process, the method aims to assign a question that maximises worker survival and expected information gain.

Different questions within a task may require knowledge and expertise on various domains. The task assignment method by Zheng et al. [166] attempts to organise questions and workers into different domains by building a knowledge base. Questions with uncertain true labels are then assigned to workers when their expertise overlap with the question’s domain.

6.2.2 Question Assignment with a Batch of Workers. Another variant of the question assignment problem is to come up with an optimal assignment scheme given a set of workers and set of questions as opposed to assigning for a sequence of workers (e.g., References [98, 171]). Cao et al. [14] termed this as the Jury Selection Problem where they aim to select a subset of crowd workers for each question under a limited budget, whose majority voting aggregated answers have the lowest probability of producing an incorrect answer.

Fan et al. [43] introduced a dynamic crowdsourcing framework named “iCrowd” that assigns tasks to workers with a higher chance of accurately completing the task using a graph-based estimation model. They consider the task similarity when estimating worker accuracy. The proposed question assignment strategy has three steps. First, it identifies a set of active workers who are ready to work on the task and dynamically finds sets of workers with the highest estimated accuracy for each available question. Then, the framework uses a greedy-approximation algorithm to formulate the optimum assignments ensuring each worker has no more than one question. Then, it strategically assigns gold-standard questions to workers who are left without any question assignments.

“AskIt,” proposed by Boim et al. [12], is another framework that achieves batchwise question assignment. The assignment method aims to minimise the global uncertainty of entropy for questions while satisfying general assignment constraints such as maximum number of answers required for each question. Two metrics are proposed to measure global uncertainty that uses the difference between maximum and minimum entropy for individual questions. AskIt uses a greedy-heuristic to come up with the optimum assignment scheme. In addition, the framework employs an initial pre-processing step that uses collaborative filtering to predict missing answers and to identify questions that are likely to be skipped by a specific worker. However, we note that the paper lacks details of the question assignment algorithm.

Goel and Faltings [51] proposed an algorithm for assigning tasks to workers that optimises the expected answer accuracy while ensuring that the collected answers satisfy pre-specified notions of error fairness. The algorithm also limits the probability of assigning many tasks to a single

worker, thus ensuring the diversity of responses. Question assignment is modelled as a constrained optimisation problem that finds the optimal crowdsourcing policy.

In a different approach, the method proposed by Li et al. [109] assigns a portion of questions to the entire worker pool and estimates the accuracy for sub-groups of workers based on characteristics such as nationality, education level and gender. Then, the framework assigns questions to workers from the specific sub-group with the highest information gain. However, this method is not practical and cost-effective when considering implementation on a crowdsourcing platform with a large number of workers from diverse backgrounds [30].

6.2.3 Blocking or Removing Workers. Question assignment can also be achieved by blocking or removing workers from the pool of eligible workers as opposed to actively assigning questions to workers. CrowdDQS [98] uses this blocking technique to further improve assignment performance. Saberi et al. [143] proposed a statistical quality control framework (OSQC) for multi-label classification tasks that monitors the performance of workers and removes workers with high error estimates at the end of processing each batch. They propose a novel method to estimate the worker accuracy—the hybrid gold plurality algorithm that uses gold-standard questions and plurality answer agreement mechanism. Question assignment is based on a Multi-rule Quality Control System that assigns a value (0,1) to the worker at the end of each batch based on the past error rate and the estimated current error rate. Early termination is also another similar strategy where workers can no longer provide answers to a particular question that already has an answer with sufficient certainty [113].

6.2.4 Question Assignment with Budget Constraints. Qiu et al. [134] investigate binary labelling tasks. Their proposed method uses previously completed gold-standard questions and estimated labels from task requesters to calculate the historic error rate for workers. Then, it predicts worker error rate for upcoming questions through an auto-regressive moving average model. Questions are assigned by maximising the accuracy with respect to the limited budget when worker payment is not constant.

Rangi and Franceschetti [135] approach task assignment with a multi-arm-bandit setup and propose using the simplified bounded KUBE algorithm as a solution. In their method, workers indicate their interest in doing the tasks, quote their charges per task, and specify the maximum number of questions they are willing to answer. Worker accuracy is estimated using the current answer distribution.

Similarly, Singer and Mittal [150] propose a pricing framework when workers bid for tasks with their expected reward and the number of questions they wish to uptake. Their method aims to maximise the number of questions completed under a fixed-budget or minimise payments for a given number of tasks.

6.2.5 Assigning Gold Standard Questions. Instead of assigning individual questions, we can also assign a specific type of question. Some frameworks have investigated whether to assign a golden-standard question or a regular question when a worker requests a task. Ipeirotis and Gabrilovich [84] presented “Quizz,” a gamified crowdsourcing system for answering multiple-choice questions. The framework uses a Markov decision process to select the next action. However, gold-standard-based question assignment alone may not lead to improved data quality due to inherent limitations in gold-standard questions discussed in Section 5.4.

6.2.6 Other Approaches. Kang and Tay [92] introduce a game-based sequential questioning strategy for question assignment in multi-class labelling questions. They convert the questions into a series of binary questions and demonstrate the reliability of their proposed approach that considers worker responses at each step.

6.3 Plurality Assignment

Crowd task accuracy can be improved by obtaining multiple answers from different workers for the same question. In a typical crowdsourcing platform, the number of answers required for each task is set by task requester prior to the task deployment. However, due to variations in worker capabilities and question difficulty [155], some questions may require more answers, whereas few answers would be sufficient for the others. Crowdsourcing research that addresses the plurality assignment problem [124] aim to dynamically decide how many answers are needed for each question.

For binary labelling tasks, Liu et al. [113] estimate the number of answers required for each question before conducting question assignment. They introduce two prediction models (basic model and an optimised version) that use workers' accuracy distribution. As such accuracy distributions are generally not available in crowdsourcing platforms, a sampling method is used to collect the accuracy of available workers.

Mo et al. [124] propose a dynamic programming-based approach to address the plurality assignment problem while maximising the output quality under a given budget. The paper identifies two key properties in crowdsourcing tasks, monotonicity and diminishing returns that describe a question with the final answer quality increasing monotonically at a decreasing rate with its plurality. They also propose an efficient greedy algorithm that can provide near optimal solutions to plurality assignment problem when monotonicity and diminishing returns properties are satisfied.

Similarly, Siddharthan et al. [149] presents an incremental Bayesian model that estimates the plurality for a classification task with a large number of categories. Results obtained through their method outperforms majority voting and is comparable to a different Bayesian approach (i.e., standard multinomial naive Bayes) that uses a larger fixed answer count.

Worker expertise and question difficulty are two key variables that impact the confidence of an answer and plurality. In a batch-processing approach, prior work by Tu et al. [155] efficiently estimated these two parameters to maximise the number of questions reliably answered at the end of each batch. In each batch, the proposed dual-cycle estimation method iteratively estimates inference between worker expertise and answer confidence, and the inference between question easiness and answer confidence in two separate cycles.

Instead of determining the plurality before task deployment, we can dynamically decide and limit the number of answers that we collect for each question. Abraham et al. [2] proposed an adaptive method that considers the differences and uncertainty of the answers provided and decide on when to stop assigning another worker for the task.

6.4 Challenges and Limitations in Task Assignment

We discuss general challenges and limitation in task assignment methods. There is no straightforward, low-cost and effective solution for task assignment [43]. Therefore, each method and evaluation has their merits and limitations.

Concerning worker accuracy estimation, some studies infer worker quality instead of objectively estimating them. For example, Saberi et al. [143] evaluate their statistical quality control framework proposed with crowd workers on Mechanical Turk where they simulate the past error rates of workers who completed the task using a standard normal distribution. Similarly, prior work by Schmitz and Lykourantzou [147] treats the work quality assessment step as a black-box process and assumes the expertise of each worker as a known numerical value. In both cases, it is difficult to argue that findings of such studies hold in real-word crowd platforms due to broader variations in crowd worker quality.

Some studies (e.g., References [6, 12, 74]) evaluate task assignment methods using synthetic data instead of using a real-time deployment or a crowdsourced dataset. Furthermore, as popular crowdsourcing platforms including Amazon Mechanical Turk do not provide sufficient means to dynamically assign tasks, all the aforementioned studies (e.g., References [43, 98, 168]) have evaluated their proposed frameworks using the external question feature of these platforms. While this is the standard for crowdsourcing research, it is unclear how worker behaviour in controlled studies compares with regular task performance.

While certain assignment methods (e.g., Reference [98]) use random or fixed values for the initial worker accuracy, other methods (e.g., References [43, 84]) use gold-standard questions. Gold-standard questions are widely used in crowdsourcing platforms. However, as discussed in Section 5, there are inherent limitations that make the use of gold questions less desirable. Also, some other methods use historic records [113] and suffer from the cold-start problem. These methods do not work with new workers in a crowdsourcing platform.

6.4.1 Heterogeneous Task Assignment Challenges. Different worker performance estimation strategies (e.g., transfer learning from similar tasks [123] and worker attributes [72, 119]) are useful for task assignment. Literature only shows that they can work on specific task types. For example, real world evaluation by Mo et al. [123] is limited to a single source and target task pair.

Overall, heterogeneous task assignment is a highly desirable approach that can potentially work across a broader range of tasks. However, more evidence and experiments are needed to show that they work with various tasks (e.g., Prior work by Hettiachchi et al. [72] uses four types of common crowdsourcing tasks) and can sustain performance over time.

6.4.2 Question Assignment Challenges. Question assignment methods continuously monitor worker answers and create assignments at each step, making them typically more effective than heterogeneous task assignment methods. However, key challenges in adopting question assignment are the complexity in implementation and the cost of calculating the assignments. For example, even with an efficient question assignment algorithm solution such as QASCA [168], assignment time linearly increase with the number of questions. Therefore, computational complexity is an important factor to consider when employing question assignment methods in a real world system.

The majority of question assignment methods are also limited to multi-class labelling problems [84, 98, 103, 168]. While literature argues that other types of tasks (e.g., a continuous value) can be converted to multi-class or binary labelling problems [168], there is no research that shows that question assignment methods can work in such cases.

6.4.3 Plurality Assignment Challenges. Plurality assignment is an important problem in crowdsourcing. Proposed methods aim to estimate plurality either upfront [113] or during the task execution [2, 155], which can help reduce the overall cost for task requesters. Similar to question assignment, estimating plurality is often investigated considering multi-class labelling questions. While it is feasible to estimate plurality for labelling questions, it is far more complicated for crowd tasks that involve complex inputs, such as audio tagging and semantic segmentation. However, plurality assignment solutions are also more valuable for such tasks as each response involves a higher work time and reward.

As plurality assignment solutions do not achieve specific worker-question match, they are less complicated than question assignment methods. Plurality assignment solutions can also be more effective when implemented together with question or task assignment methods [113]. However, further research is needed to ensure their utility in a dynamic online setting.

7 CROWDSOURCING PLATFORMS

In this section, we briefly review existing crowdsourcing platforms and standard task assignment mechanisms available in them. At a high level, current crowdsourcing platforms do not support complex task assignment methods proposed in the literature. However, certain functionalities and limited assignment methods are available to task requesters.

In Amazon Mechanical Turk,⁵ requesters can use task pre-qualifications to limit the workers who are able to see and attempt their task. The platform provides a set of pre-specified qualifications such as worker historical approval rate, location and sex. In addition, task requesters can create custom qualifications and include workers based on previous tasks or qualification tests. Further, by using MTurk API and other third-party libraries and tools (e.g., PsiTurk [64]), task requesters can build advanced task assignment methods on top of MTurk.

Toloka by Yandex⁶ is another popular crowdsourcing platform. Toloka allows task requesters to set-up worker skills that gets automatically updated based on the rate of correct responses (with gold-standard questions, majority vote, or post-verification) and behavioural features like fast responses. Requesters can also configure rules based on skills. For example, rules could automatically block workers from the task if their skill level drops below a given threshold.⁷ In addition, Toloka also provides a feature called “incremental relabeling” to facilitate dynamic plurality.

Microworkers⁸ is a similar crowdsourcing platform that provides a large collection of task templates. To facilitate basic task assignment, the platform allows custom worker groups, where requesters direct new tasks to workers who have provided satisfactory output in previous tasks. Prolific⁹ is another crowdsourcing platform that is tailored for surveys and research activities. The platform provides more than 100 demographic screeners to ensure the task is assigned for a restricted worker pool.

Other commercial crowdsourcing platforms such as Scale,¹⁰ Appen,¹¹ and Lionbridge AI¹² focus on providing an end-to-end service to task requesters. They use a combination of crowdsourced and automated approaches to complete the task. While implementation details are not available, such platforms also utilise task assignment strategies where they use automated approaches for simpler elements of the work pipeline and get crowd workers to attempt difficult parts such as quality control, edge cases, and complex data types.¹³

Further, in crowdsourcing platforms that focus on complex tasks and projects (e.g., Upwork, Freelancer, and Fiverr), task assignment is explicit. Task requesters examine the candidate workers who express willingness to complete the task and assign the task to one or more workers based on their profile. This manual assignment process is only practical for complex tasks that involve specialised workers, longer task times and higher rewards. Table 5 summarises task assignment methods offered in current commercial crowdsourcing platforms.

8 FUTURE DIRECTIONS

When discussing the future of crowd work, Kittur et al. [101] identify task assignment as one of the key elements that can improve the value and meaning of crowd work. While task assignment

⁵<https://www.mturk.com/>.

⁶Launched by Yandex in 2014. <https://toloka.ai/>.

⁷<https://toloka.ai/crowdscience/quality>.

⁸<https://www.microworkers.com/>.

⁹<https://www.prolific.co/>.

¹⁰<https://scale.com/>.

¹¹Previously Figure Eight and CrowdFlower. <https://appen.com/>.

¹²<https://lionbridge.ai/>.

¹³<https://scale.com/blog/scaling-menu-transcription-tasks-with-scale-document>.

Table 5. Task Assignment Capabilities Available in Existing Commercial Platforms

Platform	Task Assignment Methods Available
Amazon Mechanical Turk	Task pre-qualifications, Third-party integrations using the API
Appen (previously Figure 8, CrowdFlower)	Gold-standard questions
Microworkers	Assign to a custom worker group
Prolific	Demographic screeners
Toloka by Yandex	Worker skill-based task assignment, Gold-standard questions

has been increasingly researched in recent years, we do not see widespread adoption of task assignment strategies in commercial crowdsourcing platforms [23]. In this section, we reflect on limitations with current approaches and discuss how future research could address them to promote the practical use of task assignment.

One of the critical limitations of many task assignment methods is that they fail to work across a broader range of tasks. Thus, there is little incentive for crowdsourcing platforms to implement or facilitate such methods. Future work could explore more generalisable methods that do not directly depend on the task (e.g., cognitive test-based task assignment [72]). Research should also focus on how to address the cold start issue in crowdsourcing task assignment. Particularly, task requesters often do not have the luxury of collecting large volumes of training data or accessing and analysing past worker records before employing a task assignment method. Therefore, new methods that work with generic models would be more favourable to requesters.

Moreover, integrating different worker accuracy estimation methods and task assignment strategies is another feasible research direction that can further improve the value and utility of assignment methods. For example, Barbosa and Chen [9] attempt to integrate worker demographics and related attributes and show that we can improve data quality by allowing requesters to pre-specify the workforce diversity or uniformity. Similarly, research shows how cognitive [72], personality [96], and task-specific qualification tests [121] are good indicators of worker performance. Future work could investigate how to encapsulate different test scores to provide a unified estimation of worker accuracy. A prudent strategy is to implement a test marketplace, where task requesters could publish different tests that other requesters can use.

While crowdsourcing is an effective method to harness large volumes of training data for machine learning models [157], different biases (e.g., population bias, presentation bias) can be introduced through crowdsourced data collection process [120, 130]. While biases can be identified [80] and reduced in post-processing steps such as aggregation [90], future research should explore how task assignment methods can proactively manage such biases [51].

Furthermore, due to limited features and the competitive nature in crowdsourcing platforms, workers tend to use numerous third-party tools to increase their productivity [93], leading to task switching behaviour and increased fragmentation in work-life balance [162]. It is important to consider worker factors, and develop approaches that can potentially help workers manage their work (e.g., task scheduling approaches that help reduce context switching [34] and flexible ways of conducting crowd work [70]).

Finally, fair compensation for crowd workers is another important aspect [144, 161]. However, it is not sufficient to ensure that worker earnings meet the minimum hourly pay rate, requesters and platforms need to help them minimise the idle time in between jobs. In fact, task assignment reduces task search time by matching workers to compatible tasks. Future work could explore and quantify how such factors are improved through task assignment. Furthermore, assignment methods should explore task matching at a more granular level [45, 72, 98] than simply identifying

“good” or “bad” workers [142]. This will be particularly beneficial for inexperienced workers as well as others who may not be universally good at all tasks.

9 CONCLUSION

Data quality improvement methods are employed at different stages of the crowdsourcing life cycle. In this review, we provide an extensive overview of online task assignment methods in crowdsourcing that are employed during task deployment. Starting with a succinct overview of data quality improvement methods in crowdsourcing, we dissect online methods into heterogeneous task assignment, question assignment and plurality assignment problems. We discuss the challenges and limitations of existing task assignment methods, particularly their applicability, complexity, effectiveness, and cost. We anticipate that our review and discussions will help researchers and practitioners understand and adopt specific assignment methods to work for their needs. Finally, we detail a set of future research directions in crowdsourcing task assignment highlighting how research can further establish that task assignment methods are broadly applicable, beneficial to workers, and capable of mitigating biases in data.

REFERENCES

- [1] Simon à Campo, Vassilis-Javed Khan, Konstantinos Papangelis, and Panos Markopoulos. 2019. Community heuristics for user interface evaluation of crowdsourcing platforms. *Fut. Gener. Comput. Syst.* 95 (Jun. 2019), 775–789. <https://doi.org/10.1016/j.future.2018.02.028>
- [2] Ittai Abraham, Omar Alonso, Vasilis Kandyas, Rajesh Patel, Steven Sheldford, and Aleksandrs Slivkins. 2016. How many workers to ask?: Adaptive exploration for collecting high quality labels. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'16)*. ACM, 473–482. <https://doi.org/10.1145/2911451.2911514>
- [3] Harini Alagarai Sampath, Rajeev Rajeshuni, and Bipin Indurkha. 2014. Cognitively inspired task design to improve user performance on crowdsourcing platforms. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'14)*. ACM, 3665–3674. <https://doi.org/10.1145/2556288.2557155>
- [4] Omar Alonso and Ricardo Baeza-Yates. 2011. Design and implementation of relevance assessments using crowdsourcing. In *Lecture Notes in Computer Science*, Vol. 6611. Springer-Verlag, 153–164. https://doi.org/10.1007/978-3-642-20161-5_16
- [5] Vamshi Ambati, Stephan Vogel, and Jaime Carbonell. 2012. Collaborative workflow for crowdsourcing translation. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW'12)*. ACM, 1191. <https://doi.org/10.1145/2145204.2145382>
- [6] Sepehr Assadi, Justin Hsu, and Shahin Jabbari. 2015. Online assignment of heterogeneous tasks in crowdsourcing markets. In *Proceedings of the 3rd AAAI Conference on Human Computation and Crowdsourcing (HCOMP'15)*. AAAI.
- [7] Edmond Awad, Sohan Dsouza, Jean-François Bonnefon, Azim Shariff, and Iyad Rahwan. 2020. Crowdsourcing moral machines. *Commun. ACM* 63, 3 (2 2020), 48–55. <https://doi.org/10.1145/3339904>
- [8] Yukino Baba and Hisashi Kashima. 2013. Statistical quality estimation for general crowdsourcing tasks. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'13)*. ACM, 554–562. <https://doi.org/10.1145/2487575.2487600>
- [9] Natã M. Barbosa and Monchu Chen. 2019. Rehumanized crowdsourcing: A labeling framework addressing bias and ethics in machine learning. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. ACM, 1–12. <https://doi.org/10.1145/3290605.3300773>
- [10] Janine Berg. 2016. Income security in the on-demand economy: Findings and policy lessons from a survey of crowdworkers. *Compar. Labor Law Policy J.* 37, 3 (2016), 543–576.
- [11] Michael S. Bernstein, Greg Little, Robert C. Miller, Björn Hartmann, Mark S. Ackerman, David R. Karger, David Crowell, and Katrina Panovich. 2010. Soylent: A word processor with a crowd inside. In *Proceedings of the 23rd Annual ACM Symposium on User Interface Software and Technology (UIST'10)*. ACM, 313–322. <https://doi.org/10.1145/1866029.1866078>
- [12] Rubi Boim, Ohad Greenshpan, Tova Milo, Slava Novgorodov, Neoklis Polyzotis, Wang-Chiew Tan, Ohad Greenshpan, Neoklis Polyzotis, Rubi Boim, Tova Milo, and Slava Novgorodov. 2012. Asking the right questions in crowd data sourcing. In *Proceedings of the IEEE 28th International Conference on Data Engineering*. IEEE, 1261–1264. <https://doi.org/10.1109/ICDE.2012.122>

- [13] Alice M. Brawley and Cynthia L. S. Pury. 2016. Work experiences on MTurk: Job satisfaction, turnover, and information sharing. *Comput. Hum. Behav.* 54 (Jan. 2016). <https://doi.org/10.1016/j.chb.2015.08.031>
- [14] Caleb Chen Cao, Jieying She, Yongxin Tong, and Lei Chen. 2012. Whom to ask? Jury selection for decision making tasks on micro-blog services. *Proc. VLDB Endow.* 5, 11 (Jul. 2012), 1495–1506. <https://doi.org/10.14778/2350229.2350264>
- [15] L. Elisa Celis, Sai Praneeth Reddy, Ishaan Preet Singh, and Shailesh Vaya. 2016. Assignment techniques for crowdsourcing sensitive tasks. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing (CSCW'16)*. ACM, 836–847. <https://doi.org/10.1145/2818048.2835202>
- [16] Alessandro Checco, Jo Bates, and Gianluca Demartini. 2018. All that glitters is gold – An attack scheme on gold questions in crowdsourcing. In *Proceedings of the 6th AAAI Conference on Human Computation and Crowdsourcing (HCOMP'18)*. AAAI Press.
- [17] Alessandro Checco, Jo Bates, and Gianluca Demartini. 2020. Adversarial attacks on crowdsourcing quality control. *J. Artif. Intell. Res.* 67 (2020). <https://doi.org/10.1613/jair.1.11332>
- [18] Chen Chen, Xiaojun Meng, Shengdong Zhao, and Morten Fjeld. 2017. ReTool: Interactive microtask and workflow design through demonstration. In *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI'17)*. ACM, 3551–3556. <https://doi.org/10.1145/3025453.3025969>
- [19] Quanze Chen, Jonathan Bragg, Lydia B. Chilton, and Daniel S. Weld. 2019. Cicero: Multi-turn, contextual argumentation for accurate crowdsourcing. In *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI'19)*. ACM, 1–14. <https://doi.org/10.1145/3290605.3300761>
- [20] Xi Chen, Qihang Lin, and Dengyong Zhou. 2013. Optimistic knowledge gradient policy for optimal budget allocation in crowdsourcing. In *Proceedings of the 30th International Conference on Machine Learning*. 64–72.
- [21] Justin Cheng, Jaime Teevan, Shamsi T. Iqbal, and Michael S. Bernstein. 2015. Break it down: A comparison of macro- and microtasks. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI'15)*. ACM, 4061–4064. <https://doi.org/10.1145/2702123.2702146>
- [22] Peng Dai, Jeffrey M. Rzeszutarski, Praveen Paritosh, and Ed H. Chi. 2015. And now for something completely different: Improving crowdsourcing workflows with micro-diversions. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW'15)*. ACM, 628–638. <https://doi.org/10.1145/2675133.2675260>
- [23] Florian Daniel, Pavel Kucherbaev, Cinzia Cappiello, Boualem Benatallah, and Mohammad Allahbakhsh. 2018. Quality control in crowdsourcing: A survey of quality attributes, assessment techniques, and assurance actions. *Comput. Surv.* 51, 1 (Apr. 2018), 1–40. <https://doi.org/10.1145/3148148>
- [24] Akash Das Sarma, Aditya Parameswaran, and Jennifer Widom. 2016. Towards globally optimal crowdsourcing quality management. In *Proceedings of the International Conference on Management of Data (SIGMOD'16)*. ACM, 47–62. <https://doi.org/10.1145/2882903.2882953>
- [25] A. P. Dawid and A. M. Skene. 1979. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Appl. Stat.* 28, 1 (1979), 20–28. <https://doi.org/10.2307/2346806>
- [26] Gianluca Demartini, Djellel Eddine Difallah, and Philippe Cudré-Mauroux. 2012. ZenCrowd: Leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In *Proceedings of the 21st International Conference on World Wide Web (WWW'12)*. ACM, 469–478. <https://doi.org/10.1145/2187836.2187900>
- [27] Adele Diamond. 2013. Executive functions. *Annu. Rev. Psychol.* 64, 1 (2013), 135–168. <https://doi.org/10.1146/annurev-psych-113011-143750>
- [28] John P. Dickerson, Karthik Abinav Sankararaman, Aravind Srinivasan, and Pan Xu. 2018. Assigning tasks to workers based on historical data: Online task assignment with two-sided arrivals. In *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'18)*. 318–326.
- [29] Djellel Difallah, Alessandro Checco, Gianluca Demartini, and Philippe Cudré-Mauroux. 2019. Deadline-Aware fair scheduling for multi-tenant crowd-powered systems. *ACM Trans. Soc. Comput.* 2, 1 (Feb. 2019), 1–29. <https://doi.org/10.1145/3301003>
- [30] Djellel Difallah, Elena Filatova, and Panos Ipeirotis. 2018. Demographics and dynamics of mechanical turk workers. In *Proceedings of the 11th ACM International Conference on Web Search and Data Mining (WSDM'18)*. ACM, 135–143. <https://doi.org/10.1145/3159652.3159661>
- [31] Djellel Eddine Difallah, Michele Catasta, Gianluca Demartini, Panagiotis G. Ipeirotis, and Philippe Cudré-Mauroux. 2015. The dynamics of micro-task crowdsourcing: The case of amazon MTurk. In *Proceedings of the 24th International Conference on World Wide Web (WWW'15)*. IW3C2, Geneva, Switzerland, 238–247. <https://doi.org/10.1145/2736277.2741685>
- [32] Djellel Eddine Difallah, Gianluca Demartini, and Philippe Cudré-Mauroux. 2012. Mechanical cheat: Spamming schemes and adversarial techniques on crowdsourcing platforms. In *Proceedings of the CEUR Workshop*.
- [33] Djellel Eddine Difallah, Gianluca Demartini, and Philippe Cudré-Mauroux. 2013. Pick-a-crowd: Tell me what you like, and I'll tell you what to do. In *Proceedings of the 22nd International Conference on World Wide Web (WWW'13)*. ACM, 367–374. <https://doi.org/10.1145/2488388.2488421>

- [34] Djellel Eddine Difallah, Gianluca Demartini, and Philippe Cudré-Mauroux. 2016. Scheduling human intelligence tasks in multi-tenant crowd-powered systems. In *Proceedings of the 25th International Conference on World Wide Web (WWW'16)*. IW3C2, Geneva, Switzerland, 855–865. <https://doi.org/10.1145/2872427.2883030>
- [35] Evanthia Dimara, Anastasia Bezerianos, and Pierre Dragicevic. 2017. Narratives in crowdsourced evaluation of visualizations: A double-edged sword? In *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI'17)*. ACM, 5475–5484. <https://doi.org/10.1145/3025453.3025870>
- [36] Shayan Doroudi, Ece Kamar, Emma Brunskill, and Eric Horvitz. 2016. Toward a learning science for complex crowdsourcing tasks. In *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI'16)*. ACM, 2623–2634. <https://doi.org/10.1145/2858036.2858268>
- [37] Steven P. Dow, Anand Kulkarni, Scott Klemmer, and Björn Hartmann. 2012. Shepherding the crowd yields better work. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW'12)*. ACM, 1013–1022. <https://doi.org/10.1145/2145204.2145355>
- [38] Julie S. Downs, Mandy B. Holbrook, Steve Sheng, and Lorrie Faith Cranor. 2010. Are your participants gaming the system?: Screening mechanical turk workers. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'10)*. ACM, 2399–2402. <https://doi.org/10.1145/1753326.1753688>
- [39] Ryan Drapeau, Lydia B. Chilton, and Daniel S. Weld. 2016. MicroTalk: Using argumentation to improve crowdsourcing accuracy. In *Proceedings of the 4th AAAI Conference on Human Computation and Crowdsourcing (HCOMP'16)*.
- [40] Carsten Eickhoff. 2018. Cognitive biases in crowdsourcing. In *Proceedings of the 11th ACM International Conference on Web Search and Data Mining (WSDM'18)*. ACM, 162–170. <https://doi.org/10.1145/3159652.3159654>
- [41] Carsten Eickhoff, Christopher G. Harris, Arjen P. de Vries, and Padmini Srinivasan. 2012. Quality through flow and immersion: Gamifying crowdsourced relevance assessments. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'12)*. ACM, 871–880. <https://doi.org/10.1145/2348283.2348400>
- [42] Ruth B. Ekstrom, Diran Dermen, and Harry Horace Harman. 1976. *Manual for Kit of Factor-referenced Cognitive Tests*. Vol. 102. Educational Testing Service, Princeton, NJ.
- [43] Ju Fan, Guoliang Li, Beng Chin Ooi, Kian-lee Tan, and Jianhua Feng. 2015. iCrowd: An adaptive crowdsourcing framework. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD'15)*. ACM, 1015–1030. <https://doi.org/10.1145/2723372.2750550>
- [44] Ujwal Gadiraju, Alessandro Checco, Neha Gupta, and Gianluca Demartini. 2017. Modus operandi of crowd workers: The invisible role of microtask work environments. *Proc. ACM Interact. Mobile Wear. Ubiqu. Technol.* 1, 3 (9 2017), 1–29. <https://doi.org/10.1145/3130914>
- [45] Ujwal Gadiraju, Gianluca Demartini, Ricardo Kawase, and Stefan Dietze. 2019. Crowd anatomy beyond the good and bad: Behavioral traces for crowd worker modeling and pre-selection. *Comput. Support. Cooperat. Work* 28, 5 (9 2019), 815–841. <https://doi.org/10.1007/s10606-018-9336-y>
- [46] Ujwal Gadiraju, Besnik Fetahu, and Ricardo Kawase. 2015. Training workers for improving performance in Crowdsourcing Microtasks. In *Lecture Notes in Computer Science*. https://doi.org/10.1007/978-3-319-24258-3_8
- [47] Ujwal Gadiraju, Besnik Fetahu, Ricardo Kawase, Patrick Siehndel, and Stefan Dietze. 2017. Using worker self-assessments for competence-based pre-selection in crowdsourcing microtasks. *ACM Trans. Comput.-Hum. Interact.* 24, 4 (8 2017), 1–26. <https://doi.org/10.1145/3119930>
- [48] Ujwal Gadiraju, Ricardo Kawase, and Stefan Dietze. 2014. A taxonomy of microtasks on the web. In *Proceedings of the 25th ACM Conference on Hypertext and Social Media (HT'14)*. ACM, 218–223. <https://doi.org/10.1145/2631775.2631819>
- [49] Ujwal Gadiraju, Jie Yang, and Alessandro Bozzon. 2017. Clarity is a worthwhile quality: On the role of task clarity in microtask crowdsourcing. In *Proceedings of the 28th ACM Conference on Hypertext and Social Media (HT'17)*. ACM, 5–14. <https://doi.org/10.1145/3078714.3078715>
- [50] Laura Germine, Ken Nakayama, Bradley C. Duchaine, Christopher F. Chabris, Garga Chatterjee, and Jeremy B. Wilmer. 2012. Is the Web as good as the lab? Comparable performance from Web and lab in cognitive/perceptual experiments. *Psychon. Bull. Rev.* 19, 5 (10 2012), 847–857. <https://doi.org/10.3758/s13423-012-0296-9>
- [51] Naman Goel and Boi Faltings. 2019. Crowdsourcing with fairness, diversity and budget constraints. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society (AI/ES'19)*. ACM, 297–304. <https://doi.org/10.1145/3306618.3314282>
- [52] Jorge Goncalves, Michael Feldman, Subingqian Hu, Vassilis Kostakos, and Abraham Bernstein. 2017. Task routing and assignment in crowdsourcing based on cognitive abilities. In *Proceedings of the 26th International Conference on World Wide Web (WWW'17)*. IW3C2, Geneva, Switzerland, 1023–1031. <https://doi.org/10.1145/3041021.3055128>
- [53] Jorge Goncalves, Denzil Ferreira, Simo Hosio, Yong Liu, Jakob Rogstadius, Hannu Kukka, and Vassilis Kostakos. 2013. Crowdsourcing on the spot: Altruistic use of public displays, feasibility, performance, and behaviours. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing (UbiComp'13)*. ACM, 753. <https://doi.org/10.1145/2493432.2493481>

- [54] Jorge Goncalves, Simo Hosio, Denzil Ferreira, and Vassilis Kostakos. 2014. Game of words: Tagging places through crowdsourcing on public displays. In *Proceedings of the 2014 Conference on Designing Interactive Systems (DIS'14)*. ACM, 705–714. <https://doi.org/10.1145/2598510.2598514>
- [55] Jorge Goncalves, Simo Hosio, Jakob Rogstadius, Evangelos Karapanos, and Vassilis Kostakos. 2015. Motivating participation and improving quality of contribution in ubiquitous crowdsourcing. *Comput. Netw.* 90 (Oct. 2015), 34–48. <https://doi.org/10.1016/j.comnet.2015.07.002>
- [56] Jorge Goncalves, Simo Hosio, Niels van Berkel, Furqan Ahmed, and Vassilis Kostakos. 2017. CrowdPickUp. *Proc. ACM Interact. Mobile Wear. Ubiq. Technol.* 1, 3 (Sep. 2017), 1–22. <https://doi.org/10.1145/3130916>
- [57] Jorge Goncalves, Vassilis Kostakos, Simo Hosio, Evangelos Karapanos, and Olga Lyra. 2013. IncluCity: Using contextual cues to raise awareness on environmental accessibility. In *Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility*. ACM, 1–8. <https://doi.org/10.1145/2513383.2517030>
- [58] Jorge Goncalves, Hannu Kukka, Iván Sánchez, and Vassilis Kostakos. 2016. Crowdsourcing queue estimations in situ. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing (CSCW'16)*. ACM, 1040–1051. <https://doi.org/10.1145/2818048.2819997>
- [59] Jorge Goncalves, Pratyush Pandab, Denzil Ferreira, Mohammad Ghahramani, Guoying Zhao, and Vassilis Kostakos. 2014. Projective testing of diurnal collective emotion. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp'14)*. 487–497. <https://doi.org/10.1145/2632048.2636067>
- [60] Tanya Goyal, Tyler McDonnell, Mucahid Kutlu, Tamer Elsayed, and Matthew Lease. 2018. Your behavior signals your reliability: Modeling crowd behavioral traces to ensure quality relevance annotations. In *Proceedings of the 6th AAAI Conference on Human Computation and Crowdsourcing (HCOMP'18)*. AAAI Press, 41–49.
- [61] Srinivasa Raghavendra Bhuvan Gummididi, Xike Xie, and Torben Bach Pedersen. 2019. A survey of spatial crowdsourcing. *ACM Trans. Datab. Syst.* 44, 2, Article 8 (2019), 1–46. <https://doi.org/10.1145/3291933>
- [62] Bin Guo, Zhu Wang, Zhiwen Yu, Yu Wang, Neil Y. Yen, Runhe Huang, and Xingshe Zhou. 2015. Mobile crowd sensing and computing: The review of an emerging human-powered sensing paradigm. *ACM Comput. Surv.* 48, 1, Article 7 (Aug. 2015), 31 pages. <https://doi.org/10.1145/2794400>
- [63] Stephen Guo, Aditya Parameswaran, and Hector Garcia-Molina. 2012. So who won? Dynamic max discovery with the crowd. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD'12)*. ACM, 385–396. <https://doi.org/10.1145/2213836.2213880>
- [64] Todd M. Gureckis, Jay Martin, John McDonnell, Alexander S. Rich, Doug Markant, Anna Coenen, David Halpern, Jessica B. Hamrick, and Patricia Chan. 2016. psiTurk: An open-source framework for conducting replicable behavioral experiments online. *Behav. Res. Methods* 48, 3 (Sep. 2016), 829–842. <https://doi.org/10.3758/s13428-015-0642-8>
- [65] Shuguang Han, Peng Dai, Praveen Paritosh, and David Huynh. 2016. Crowdsourcing human annotation on web page structure. *ACM Trans. Intell. Syst. Technol.* 7, 4 (Apr. 2016), 1–25. <https://doi.org/10.1145/2870649>
- [66] Anikó Hannák, Claudia Wagner, David Garcia, Alan Mislove, Markus Strohmaier, and Christo Wilson. 2017. Bias in online freelance marketplaces: Evidence from taskrabit and fiverr. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work and Social Computing (CSCW'17)*. ACM, 1914–1933. <https://doi.org/10.1145/2998181.2998327>
- [67] Kotaro Hara, Vicki Le, and Jon Froehlich. 2013. Combining crowdsourcing and google street view to identify street-level accessibility problems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'13)*. ACM, 631–640. <https://doi.org/10.1145/2470654.2470744>
- [68] Alireza Hassani, Pari Delir Haghighi, and Prem Prakash Jayaraman. 2015. Context-Aware recruitment scheme for opportunistic mobile crowdsensing. In *Proceedings of the IEEE 21st International Conference on Parallel and Distributed Systems (ICPADS'15)*. IEEE, 266–273. <https://doi.org/10.1109/ICPADS.2015.41>
- [69] Danula Hettiachchi and Jorge Goncalves. 2019. Towards effective crowd-powered online content moderation. In *Proceedings of the 31st Australian Conference on Human-Computer-Interaction*. ACM, 342–346. <https://doi.org/10.1145/3369457.3369491>
- [70] Danula Hettiachchi, Zhanna Sarsenbayeva, Fraser Allison, Niels van Berkel, Tilman Dingler, Gabriele Marini, Vassilis Kostakos, and Jorge Goncalves. 2020. “Hi! I am the Crowd Tasker” Crowdsourcing through Digital Voice Assistants. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI'20)*. ACM. <https://doi.org/10.1145/3313831.3376320>
- [71] Danula Hettiachchi, Niels van Berkel, Simo Hosio, Vassilis Kostakos, and Jorge Goncalves. 2019. Effect of cognitive abilities on crowdsourcing task performance. In *Proceedings of the IFIP TC13 International Conference on Human-Computer Interaction (INTERACT'19)*. Springer International Publishing, Cham, 442–464. https://doi.org/10.1007/978-3-030-29381-9_28
- [72] Danula Hettiachchi, Niels van Berkel, Vassilis Kostakos, and Jorge Goncalves. 2020. CrowdCog: A cognitive skill based system for heterogeneous task assignment and recommendation in crowdsourcing. *Proc. ACM Hum.-Comput. Interact.* 4, 2 (Oct. 2020), 1–22. <https://doi.org/10.1145/3415181>

- [73] Danula Hettiachchi, Senuri Wijenayake, Simo Hosio, Vassilis Kostakos, and Jorge Goncalves. 2020. How context influences cross-device task acceptance in crowd work. In *Proceedings of the 8th AAAI Conference on Human Computation and Crowdsourcing (HCOMP'20)*. AAAI Press, 53–62.
- [74] Chien Ju Ho, Shahin Jabbari, and Jennifer Wortman Vaughan. 2013. Adaptive task assignment for crowdsourced classification. In *Proceedings of the 30th International Conference on Machine Learning (ICML'13)*, 534–542.
- [75] Chien Ju Ho and Jennifer Wortman Vaughan. 2012. Online task assignment in crowdsourcing markets. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence Online*, Vol. 1. 45–51.
- [76] Simo Hosio, Jorge Goncalves, Vassilis Kostakos, and Jukka Rieki. 2015. Crowdsourcing public opinion using urban pervasive technologies: Lessons from real-life experiments in Oulu. *Policy Internet* 7, 2 (2015), 203–222. <https://doi.org/10.1002/poi3.90>
- [77] Simo Hosio, Jorge Goncalves, Vili Lehdonvirta, Denzil Ferreira, and Vassilis Kostakos. 2014. Situated crowdsourcing using a market model. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST'14)*. ACM, 55–64. <https://doi.org/10.1145/2642918.2647362>
- [78] Simo Johannes Hosio, Jaro Karppinen, Esa-Pekka Takala, Jani Takatalo, Jorge Goncalves, Niels van Berkel, Shin'ichi Konomi, and Vassilis Kostakos. 2018. Crowdsourcing treatments for low back pain. In *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI'18)*. ACM, 1–12. <https://doi.org/10.1145/3173574.3173850>
- [79] Jeff Howe. 2006. The rise of crowdsourcing. *Wired Mag.* (2006). <https://doi.org/10.1086/599595>
- [80] Xiao Hu, Haobo Wang, Anirudh Vegesana, Somesh Dube, Kaiwen Yu, Gore Kao, Shuo-Han Chen, Yung-Hsiang Lu, George K. Thiruvathukal, and Ming Yin. 2020. Crowdsourcing detection of sampling biases in image datasets. In *Proceedings of the International Conference on World Wide Web (WWW'20)*. ACM, 2955–2961. <https://doi.org/10.1145/3366423.3380063>
- [81] Nguyen Quoc Viet Hung, Duong Chi Thang, Matthias Weidlich, and Karl Aberer. 2015. Minimizing efforts in validating crowd answers. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD'15)*. ACM, 999–1014. <https://doi.org/10.1145/2723372.2723731>
- [82] Kazushi Ikeda and Keiichiro Hoashi. 2017. Crowdsourcing GO: Effect of worker situation on mobile crowdsourcing performance. In *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI'17)*. ACM, 1142–1153. <https://doi.org/10.1145/3025453.3025917>
- [83] Kosetsu Ikeda, Atsuyuki Morishima, Habibur Rahman, Senjuti Basu Roy, Saravanan Thirumuruganathan, Sihem Amer-Yahia, and Gautam Das. 2016. Collaborative crowdsourcing with Crowd4U. *Proc. VLDB Endow.* 9, 13 (Sep. 2016), 1497–1500. <https://doi.org/10.14778/3007263.3007293>
- [84] Panagiotis G. Ipeirotis and Evgeniy Gabrilovich. 2014. Quiz: Targeted crowdsourcing with a billion (potential) users. In *Proceedings of the 23rd International Conference on World Wide Web (WWW'14)*. ACM, 143–154.
- [85] Lilly C. Irani and M. Six Silberman. 2013. Turkopticon: Interrupting worker invisibility in Amazon Mechanical Turk. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'13)*. ACM, 611–620. <https://doi.org/10.1145/2470654.2470742>
- [86] Manas Joglekar, Hector Garcia-Molina, and Aditya Parameswaran. 2015. Comprehensive and reliable crowd assessment algorithms. In *Proceedings of the International Conference on Data Engineering*. IEEE, 195–206. <https://doi.org/10.1109/ICDE.2015.7113284>
- [87] Oliver P. John, Laura P. Naumann, and Christopher J. Soto. 2008. Paradigm shift to the integrative big five trait taxonomy. *Handb. Personal.: Theory Res.* 3, 2 (2008), 114–158.
- [88] Hyun Joon Jung and Matthew Lease. 2011. Improving consensus accuracy via Z-score and weighted voting. In *AAAI Workshop—Technical Report*.
- [89] Sanjay Kairam and Jeffrey Heer. 2016. Parting crowds: Characterizing divergent interpretations in crowdsourced annotation tasks. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing (CSCW'16)*. ACM, 1635–1646. <https://doi.org/10.1145/2818048.2820016>
- [90] Ece Kamar, Ashish Kapoor, and Eric Horvitz. 2015. Identifying and accounting for task-dependent bias in crowdsourcing. In *Proceedings of the 3rd AAAI Conference on Human Computation and Crowdsourcing (HCOMP'15)*.
- [91] Georges A. Kamhoua, Niki Pissinou, S. S. Iyengar, Jonathan Beltran, Jerry Miller, Charles A. Kamhoua, and Laurent L. Njilla. 2018. Approach to detect non-adversarial overlapping collusion in crowdsourcing. In *Proceedings of the IEEE 36th International Performance Computing and Communications Conference (IPCCC'17)*. <https://doi.org/10.1109/IPCCC.2017.8280462>
- [92] Qiyu Kang and Wee Peng Tay. 2017. Sequential multi-class labeling in crowdsourcing: A ulam-renyi game approach. In *Proceedings of the International Conference on Web Intelligence (WI'17)*. ACM, 245–251. <https://doi.org/10.1145/3106426.3106446>
- [93] Toni Kaplan, Susumu Saito, Kotaro Hara, and Jeffrey Bigham. 2018. Striving to earn more: A survey of work strategies and tool use among crowd workers. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing (HCOMP'18)*.

- [94] David R. Karger, Sewoong Oh, and Devavrat Shah. 2013. Efficient crowdsourcing for multi-class labeling. *Perf. Eval. Rev.* 41, 1 (2013), 81–92. <https://doi.org/10.1145/2494232.2465761>
- [95] Gabriella Kazai, Jaap Kamps, and Natasa Milic-Frayling. 2011. Worker types and personality traits in crowdsourcing relevance labels. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management (CIKM'11)*. ACM, 1941–1944. <https://doi.org/10.1145/2063576.2063860>
- [96] Gabriella Kazai, Jaap Kamps, and Natasa Milic-Frayling. 2012. The face of quality in crowdsourcing relevance labels. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM'12)*. ACM, 2583. <https://doi.org/10.1145/2396761.2398697>
- [97] Gabriella Kazai and Imed Zitouni. 2016. Quality management in crowdsourcing using gold judges behavior. In *Proceedings of the 9th ACM International Conference on Web Search and Data Mining (WSDM'16)*. ACM, 267–276. <https://doi.org/10.1145/2835776.2835835>
- [98] Asif R. Khan and Hector Garcia-Molina. 2017. CrowdDQS: Dynamic question selection in crowdsourcing systems. In *Proceedings of the ACM International Conference on Management of Data (SIGMOD'17)*. ACM, 1447–1462. <https://doi.org/10.1145/3035918.3064055>
- [99] Ashiqur R. KhudaBukhsh, Jaime G. Carbonell, and Peter J Jansen. 2014. Detecting non-adversarial collusion in crowdsourcing. In *Proceedings of the 2nd AAAI Conference on Human Computation and Crowdsourcing (HCOMP'14)*. AAAI Press.
- [100] Aniket Kittur, Susheel Khamkar, Paul André, and Robert Kraut. 2012. CrowdWeaver: Visually managing complex crowd work. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work (CSCW'12)*. ACM, 1033. <https://doi.org/10.1145/2145204.2145357>
- [101] Aniket Kittur, Jeffrey V. Nickerson, Michael Bernstein, Elizabeth Gerber, Aaron Shaw, John Zimmerman, Matthew Lease, and John Horton. 2013. The future of crowd work. In *Proceedings of the 2013 Conference on Computer Supported Cooperative Work (CSCW'13)*. ACM, 1301. <https://doi.org/10.1145/2441776.2441923>
- [102] Aniket Kittur, Boris Smus, Susheel Khamkar, and Robert E. Kraut. 2011. CrowdForge: Crowdsourcing complex work. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST'11)*. ACM, 43–52. <https://doi.org/10.1145/2047196.2047202>
- [103] Ari Kobren, Chun How Tan, Panagiotis G. Ipeirotis, and Evgeniy Gabrilovich. 2015. Getting more for less: Optimized crowdsourcing with dynamic tasks and goals. In *Proceedings of the 24th International Conference on World Wide Web (WWW'15)*. ACM, USA, 592–602. <https://doi.org/10.1145/2736277.2741681>
- [104] Li Kuang, Huan Zhang, Ruyi Shi, Zhifang Liao, and Xiaoxian Yang. 2020. A spam worker detection approach based on heterogeneous network embedding in crowdsourcing platforms. *Comput. Netw.* 183 (Dec. 2020), 107587. <https://doi.org/10.1016/j.comnet.2020.107587>
- [105] Anand Kulkarni, Matthew Can, and Björn Hartmann. 2012. Collaboratively crowdsourcing workflows with turkomatic. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work (CSCW'12)*. ACM, 1003–1012. <https://doi.org/10.1145/2145204.2145354>
- [106] Katsumi Kumai, Masaki Matsubara, Yuhki Shiraishi, Daisuke Wakatsuki, Jianwei Zhang, Takeaki Shionome, Hiroyuki Kitagawa, and Atsuyuki Morishima. 2018. Skill-and-Stress-Aware assignment of crowd-worker groups to task streams. In *Proceedings of the 6th AAAI Conference on Human Computation and Crowdsourcing (HCOMP'18)*. AAAI Press, 88–97.
- [107] John Le, Andy Edmonds, Vaughn Hester, and Lukas Biewald. 2010. Ensuring quality in crowdsourced search relevance evaluation: The effects of training question distribution. In *Proceedings of the SIGIR 2010 Workshop on Crowdsourcing for Search Evaluation*.
- [108] Guoliang Li, Jiannan Wang, Yudian Zheng, and Michael J. Franklin. 2016. Crowdsourced data management: A survey. *IEEE Trans. Knowl. Data Eng.* 28, 9 (9 2016), 2296–2319. <https://doi.org/10.1109/TKDE.2016.2535242>
- [109] Hongwei Li, Bo Zhao, and Ariel Fuxman. 2014. The wisdom of minority: Discovering and targeting the right group of workers for crowdsourcing. In *Proceedings of the 23rd International Conference on World Wide Web (WWW'14)*. ACM, 165–176. <https://doi.org/10.1145/2566486.2568033>
- [110] Greg Little, Lydia B. Chilton, Max Goldman, and Robert C. Miller. 2010. Exploring iterative and parallel human computation processes. In *Proceedings of the Human Computation Workshop (HCOMP'10)*. <https://doi.org/10.1145/1837885.1837907>
- [111] Qiang Liu, Alexander T. Ihler, and Mark Steyvers. 2013. Scoring workers in crowdsourcing: How many control questions are enough?. In *Advances in Neural Information Processing Systems*, Vol. 26. 1914–1922.
- [112] Qiang Liu, Jian Peng, and Alexander T. Ihler. 2012. Variational inference for crowdsourcing. In *Advances in Neural Information Processing Systems*, Vol. 25. 692–700.
- [113] Xuan Liu, Meiyu Lu, Beng Chin Ooi, Yanyan Shen, Sai Wu, and Meihui Zhang. 2012. CDAS: A crowdsourcing data analytics system. *Proc. VLDB Endow.* 5, 10 (Jun. 2012), 1040–1051. <https://doi.org/10.14778/2336664.2336676>

- [114] Joanna Lykourentzou, Angeliki Antoniou, Yannick Naudet, and Steven P. Dow. 2016. Personality matters: Balancing for personality types leads to better outcomes for crowd teams. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing (CSCW'16)*. ACM, 260–273. <https://doi.org/10.1145/2818048.2819979>
- [115] Fenglong Ma, Yaliang Li, Qi Li, Minghui Qiu, Jing Gao, Shi Zhi, Lu Su, Bo Zhao, Heng Ji, and Jiawei Han. 2015. FaitCrowd: Fine grained truth discovery for crowdsourced data aggregation. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'15)*. ACM, 745–754. <https://doi.org/10.1145/2783258.2783314>
- [116] V. K. Chaithanya Manam and Alexander J. Quinn. 2018. WingIt: Efficient refinement of unclear task instructions. In *Proceedings of the 6th AAI Conference on Human Computation and Crowdsourcing (HCOMP'18, Vol. 6)*. AAAI Press.
- [117] Andrew Mao, Ece Kamar, Yiling Chen, Eric Horvitz, Megan E. Schwamb, Chris J. Lintott, and Arfon M. Smith. 2013. Volunteering versus work for pay: Incentives and tradeoffs in crowdsourcing. In *Proceedings of the AAI Conference on Human Computation and Crowdsourcing (HCOMP'13), Vol. 1*. AAAI Press.
- [118] William Moulton Marston. 2013. *Emotions of Normal People*. Vol. 158. Routledge.
- [119] Panagiotis Mavridis, David Gross-Amblard, and Zoltán Miklós. 2016. Using hierarchical skills for optimized task assignment in knowledge-intensive crowdsourcing. In *Proceedings of the 25th International Conference on World Wide Web (WWW'16)*. IW3C2, Geneva, Switzerland, 843–853. <https://doi.org/10.1145/2872427.2883070>
- [120] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2021. A survey on bias and fairness in machine learning. *ACM Comput. Surv.* 54, 6, Article 115 (Jul. 2021), 35 pages. <https://doi.org/10.1145/3457607>
- [121] Tanushree Mitra, C. J. Hutto, and Eric Gilbert. 2015. Comparing person- and process-centric strategies for obtaining quality data on amazon mechanical turk. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI'15)*. ACM, 1345–1354. <https://doi.org/10.1145/2702123.2702553>
- [122] Ken Mizusawa, Keishi Tajima, Masaki Matsubara, Toshiyuki Amagasa, and Atsuyuki Morishima. 2018. Efficient pipeline processing of crowdsourcing workflows. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM'18)*. ACM, 1559–1562. <https://doi.org/10.1145/3269206.3269292>
- [123] Kaixiang Mo, Erheng Zhong, and Qiang Yang. 2013. Cross-task crowdsourcing. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'13)*. ACM, 677–685. <https://doi.org/10.1145/2487575.2487593>
- [124] Luyi Mo, Reynold Cheng, Ben Kao, Xuan S. Yang, Chenghui Ren, Siyu Lei, David W. Cheung, and Eric Lo. 2013. Optimizing plurality for human intelligence tasks. In *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management (CIKM'13)*. ACM, 1929–1938. <https://doi.org/10.1145/2505515.2505755>
- [125] Benedikt Morschheuser, Juho Hamari, and Jonna Koivisto. 2016. Gamification in crowdsourcing: A review. In *Proceedings of the 49th Hawaii International Conference on System Sciences (HICSS'16)*. IEEE, 4375–4384. <https://doi.org/10.1109/HICSS.2016.543>
- [126] Yashar Moshfeghi, Alvaro F. Huertas-Rosero, and Joemon M. Jose. 2016. Identifying careless workers in crowdsourcing platforms. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'16)*. ACM, 857–860. <https://doi.org/10.1145/2911451.2914756>
- [127] Mohamed Musthag and Deepak Ganesan. 2013. Labor dynamics in a mobile micro-task market. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'13)*. ACM, 641–650. <https://doi.org/10.1145/2470654.2470745>
- [128] Stefanie Nowak and Stefan R uger. 2010. How reliable are annotations via crowdsourcing. In *Proceedings of the International Conference on Multimedia Information Retrieval (MIR'10)*. ACM, 557. <https://doi.org/10.1145/1743384.1743478>
- [129] David Oleson, Alexander Sorokin, Greg Laughlin, Vaughn Hester, John Le, and Lukas Biewald. 2011. Programmatic gold: Targeted and scalable quality assurance in crowdsourcing. In *AAAI Workshop—Technical Report*.
- [130] Alexandra Olteanu, Carlos Castillo, Fernando Diaz, and Emre Kiciman. 2019. Social data: Biases, methodological pitfalls, and ethical boundaries. *Front. Big Data* 2, 13 (2019). <https://doi.org/10.3389/fdata.2019.00013>
- [131] Dim P. Papadopoulos, Jasper R. R. Uijlings, Frank Keller, and Vittorio Ferrari. 2017. Extreme clicking for efficient object annotation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV'17)*. 4940–4949. <https://doi.org/10.1109/ICCV.2017.528>
- [132] Sunghyun Park, Philippa Shoemark, and Louis-Philippe Morency. 2014. Toward crowdsourcing micro-level behavior annotations: The challenges of interface, training, and generalization. In *Proceedings of the 19th International Conference on Intelligent User Interfaces (IUI'14)*. ACM, 37–46. <https://doi.org/10.1145/2557500.2557512>
- [133] Eyal Peer, Joachim Vosgerau, and Alessandro Acquisti. 2014. Reputation as a sufficient condition for data quality on Amazon Mechanical Turk. *Behav. Res. Methods* 46, 4 (Dec. 2014), 1023–1031. <https://doi.org/10.3758/s13428-013-0434-y>

- [134] Chenxi Qiu, Anna C. Squicciarini, Barbara Carminati, James Caverlee, and Dev Rishi Khare. 2016. CrowdSelect: Increasing accuracy of crowdsourcing tasks through behavior prediction and user selection. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management (CIKM'16)*. ACM, 539–548. <https://doi.org/10.1145/2983323.2983830>
- [135] Anshuka Rangi and Massimo Franceschetti. 2018. Multi-armed bandit algorithms for crowdsourcing systems with online estimation of workers' ability. In *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'18)*. 1345–1352.
- [136] Vikas C. Raykar and Shipeng Yu. 2012. Eliminating spammers and ranking annotators for crowdsourced labeling tasks. *J. Mach. Learn. Res.* 13 (Feb. 2012), 491–518.
- [137] Vikas C. Raykar, Shipeng Yu, Linda H. Zhao, Anna Jerebko, Charles Florin, Gerardo Hermosillo Valadez, Luca Bogoni, and Linda Moy. 2009. Supervised learning from multiple experts: Whom to trust when everyone lies a bit. In *Proceedings of the 26th International Conference on Machine Learning (ICML'09)*.
- [138] Daniela Retelny, Michael S. Bernstein, and Melissa A. Valentine. 2017. No workflow can ever be enough. *Proc. ACM Hum.-Comput. Interact.* 1 (Dec. 2017), 1–23. <https://doi.org/10.1145/3134724>
- [139] Mirela Riveni, Tien-Dung Nguyen, Mehmet S. Aktas, and Schahram Dustdar. 2019. Application of provenance in social computing: A case study. *Concurr. Comput.: Pract. Exper.* 31, 3 (2019), e4894. <https://doi.org/10.1002/cpe.4894>
- [140] Jakob Rogstadius, Vassilis Kostakos, Aniket Kittur, Boris Smus, Jim Laredo, and Maja Vukovic. 2011. An assessment of intrinsic and extrinsic motivation on task performance in crowdsourcing markets. In *Proceedings of the 5th International AAAI Conference on Web and Social Media (ICWSM'11)*. AAAI, 17–21.
- [141] Joel Ross and Bill Tomlinson. 2010. Who are the crowdworkers? Shifting demographics in mechanical turk. In *CHI'10 Extended Abstracts on Human Factors in Computing Systems*. 2863–2872.
- [142] Jeffrey M. Rzeszotarski and Aniket Kittur. 2011. Instrumenting the crowd: Using implicit behavioral measures to predict task performance. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST'11)*. ACM, 13–22. <https://doi.org/10.1145/2047196.2047199>
- [143] Morteza Saberi, Omar K. Hussain, and Elizabeth Chang. 2017. An online statistical quality control framework for performance management in crowdsourcing. In *Proceedings of the International Conference on Web Intelligence (WI'17)*. ACM, 476–482. <https://doi.org/10.1145/3106426.3106436>
- [144] Niloufar Salehi, Lilly C. Irani, Michael S. Bernstein, Ali Alkhatib, Eva Ogbe, Kristy Milland, and Clickhappier. 2015. We are dynamo: Overcoming stalling and friction in collective action for crowd workers. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI'15)*. ACM, 1621–1630. <https://doi.org/10.1145/2702123.2702508>
- [145] Mike Schaeckermann, G. O. H. Joslin, Kate Larson, and L. A. W. Edith. 2018. Resolvable vs. Irresolvable disagreement: A study on worker deliberation in crowd work. *Proc. ACM Hum.-Comput. Interact.* 2, CSCW, Article 154 (2018), 1–19. <https://doi.org/10.1145/3274423>
- [146] Daniel Schall, Florian Skopik, and Schahram Dustdar. 2012. Expert discovery and interactions in mixed service-oriented systems. *IEEE Trans. Serv. Comput.* 5, 2 (2012), 233–245. <https://doi.org/10.1109/TSC.2011.2>
- [147] Heinz Schmitz and Ioanna Lykourantzou. 2018. Online sequencing of non-decomposable macrotasks in expert crowdsourcing. *ACM Trans. Soc. Comput.* 1, 1 (2018), 1–33. <https://doi.org/10.1145/3140459>
- [148] Aaron D. Shaw, John J. Horton, and Daniel L. Chen. 2011. Designing incentives for inexperienced human raters. In *Proceedings of the ACM 2011 Conference on Computer Supported Cooperative Work (CSCW'11)*. ACM, 275. <https://doi.org/10.1145/1958824.1958865>
- [149] Advait Siddharthan, Christopher Lambin, Anne Marie Robinson, Nirwan Sharma, Richard Comont, Elaine O'mahony, Chris Mellish, and René Van Der Wal. 2016. Crowdsourcing without a crowd: Reliable online species identification using Bayesian models to minimize crowd size. *ACM Trans. Intell. Syst. Technol.* 7, 4 (2016). <https://doi.org/10.1145/2776896>
- [150] Yaron Singer and Manas Mittal. 2013. Pricing mechanisms for crowdsourcing markets. In *Proceedings of the 22nd International Conference on World Wide Web (WWW'13)*. ACM, 1157–1166. <https://doi.org/10.1145/2488388.2488489>
- [151] Klaas Jan Stol and Brian Fitzgerald. 2014. Two's company, three's a crowd: A case study of crowdsourcing software development. In *Proceedings of the International Conference on Software Engineering*. IEEE Computer Society, 187–198. <https://doi.org/10.1145/2568225.2568249>
- [152] Hao Su, Jia Deng, and Li Fei-Fei. 2012. Crowdsourcing annotations for visual object detection. In *AAAI Workshop—Technical Report*.
- [153] Yongxin Tong, Zimu Zhou, Yuxiang Zeng, Lei Chen, and Cyrus Shahabi. 2020. Spatial crowdsourcing: A survey. *The VLDB J.* 29, 1 (1 2020), 217–250. <https://doi.org/10.1007/s00778-019-00568-7>
- [154] Long Tran-Thanh, Trung Dong Huynh, Avi Rosenfeld, Sarvapali D. Ramchurn, and Nicholas R. Jennings. 2015. Crowdsourcing complex workflows under budget constraints. In *Proceedings of the National Conference on Artificial Intelligence (AAAI'15)*. AAAI Press, 1298–1304.

- [155] Jiayang Tu, Peng Cheng, and Lei Chen. 2019. Quality-Assured synchronized task assignment in crowdsourcing. *IEEE Trans. Knowl. Data Eng.* 4347, c (2019), 1–1. <https://doi.org/10.1109/tkde.2019.2935443>
- [156] Aditya Vashistha, Pooja Sethi, and Richard Anderson. 2017. Respeak: A voice-based, crowd-powered speech transcription system. In *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI'17)*. ACM, 1855–1866. <https://doi.org/10.1145/3025453.3025640>
- [157] Jennifer Wortman Vaughan. 2017. Making better use of the crowd: How crowdsourcing can advance machine learning research. *J. Mach. Learn. Res.* 18, 1 (2017), 7026–7071.
- [158] Matteo Venanzi, John Guiver, Gabriella Kazai, Pushmeet Kohli, and Milad Shokouhi. 2014. Community-based Bayesian aggregation models for crowdsourcing. In *Proceedings of the 23rd International Conference on World Wide Web (WWW'14)*. ACM, 155–164. <https://doi.org/10.1145/2566486.2567989>
- [159] Jacob Whitehill, Ting-fan Wu, Jacob Bergsma, Javier Movellan, and Paul Ruvolo. 2009. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *Advances in Neural Information Processing Systems*, Vol. 22.
- [160] Mark E. Whiting, Dilrukshi Gamage, Snehal Kumar (Neil) S. Gaikwad, Aaron Gilbee, Shirish Goyal, and Others. 2017. Crowd guilds: Worker-Led reputation and feedback on crowdsourcing platforms. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work and Social Computing (CSCW'17)*. ACM, 1902–1913. <https://doi.org/10.1145/2998181.2998234>
- [161] Mark E. Whiting, Grant Hugh, and Michael S. Bernstein. 2019. Fair work: Crowd work minimum wage with one line of code. In *Proceedings of the 7th AAAI Conference on Human Computation and Crowdsourcing (HCOMP'19)*. 197–206.
- [162] Alex C. Williams, Gloria Mark, Kristy Milland, Edward Lank, and Edith Law. 2019. The perpetual work life of crowdworkers: How tooling practices increase fragmentation in crowdwork. *Proc. ACM Hum.-Comput. Interact.* 3, (2019). <https://doi.org/10.1145/3359126>
- [163] Dong Yuan, Guoliang Li, Qi Li, and Yudian Zheng. 2017. Sybil defense in crowdsourcing platforms. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (CIKM'17)*. ACM, 1529–1538. <https://doi.org/10.1145/3132847.3133039>
- [164] Jing Zhang, Xindong Wu, and Victor S. Sheng. 2016. Learning from crowdsourced labeled data: A survey. *Artif. Intell. Rev.* 46, 4 (12 2016), 543–576. <https://doi.org/10.1007/s10462-016-9491-9>
- [165] Zhou Zhao, James Cheng, Furu Wei, Ming Zhou, Wilfred Ng, and Yingjun Wu. 2014. SocialTransfer: Transferring social knowledge for cold-start crowdsourcing. In *Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM'14)*. 779–788. <https://doi.org/10.1145/2661829.2661871>
- [166] Yudian Zheng, Guoliang Li, and Reynold Cheng. 2016. DOCS: Domain-aware crowdsourcing system using knowledge bases. *Proc. VLDB Endow.* 10, 4 (2016), 361–372. <https://doi.org/10.14778/3025111.3025118>
- [167] Yudian Zheng, Guoliang Li, Yuanbing Li, Caihua Shan, and Reynold Cheng. 2017. Truth inference in crowdsourcing. *Proc. VLDB Endow.* 10, 5 (1 2017), 541–552. <https://doi.org/10.14778/3055540.3055547>
- [168] Yudian Zheng, Jiannan Wang, Guoliang Li, Reynold Cheng, and Jianhua Feng. 2015. QASCA: A quality-aware task assignment system for crowdsourcing applications. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD'15)*. ACM, 1031–1046. <https://doi.org/10.1145/2723372.2749430>
- [169] Dengyong Zhou, Sumit Basu, Yi Mao, and John Platt. 2012. Learning from the wisdom of crowds by minimax entropy. In *Advances in Neural Information Processing Systems*, Vol. 25.
- [170] Haiyi Zhu, Steven P. Dow, Robert E. Kraut, and Aniket Kittur. 2014. Reviewing versus doing: Learning and performance in crowd assessment. In *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW'14)*. ACM, 1445–1455. <https://doi.org/10.1145/2531602.2531718>
- [171] Honglei Zhuang, Aditya Parameswaran, Dan Roth, and Jiawei Han. 2015. Debiasing crowdsourced batches. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'15)*. ACM, 1593–1602. <https://doi.org/10.1145/2783258.2783316>
- [172] Mengdie Zhuang and Ujwal Gadiraju. 2019. In what mood are you today? An analysis of crowd workers' mood, performance and engagement mengdie. In *Proceedings of the 10th ACM Conference on Web Science (WebSci'19)*. ACM, 373–382. <https://doi.org/10.1145/3292522.3326010>

Received February 2021; revised October 2021; accepted October 2021