Interaction Management in Crowdsourcing

Yansheng Wang [†], Tianshu Song [†], Qian Tao [†], Yuxiang Zeng [‡], Zimu Zhou [#],
Yi Xu [†], Yongxin Tong [†], Lei Chen [‡]
[†]BDBC, SKLSDE Lab and IRI, Beihang University, China
[‡]The Hong Kong University of Science and Technology, Hong Kong SAR, China
[#] Singapore Management University, Singapore
[†]{arthur_wang, songts, qiantao, xuy, yxtong}@buaa.edu.cn
[‡]{yzengal, leichen}@cse.ust.hk, [#]zimuzhou@smu.edu.sg

Abstract

Crowdsourcing is anticipated as a promising paradigm for Future of Work (FoW), where groups of humans are engaged for problem-solving, services and innovation which are usually difficult for machines. During the entire workflow of crowdsourcing, intensive interactions take place between workers and the crowdsourcing platform, as well as among groups of workers. For sustainable crowdsourcing, the design and management of these interactions should not regard human workers as machines, but rather as individuals and social beings. In this article, we highlight the critical interactions in typical crowdsourcing ecosystems, summarize past efforts on human-centric interaction management in crowdsourcing, and discuss emerging interaction management research towards cross-platform crowdsourcing.

1 Introduction

Some computation tasks are intractable for computers (or machines) but easy for human beings, such as image recognition, text translation, entity resolution, etc. Although the new breakthroughs of artificial intelligence and machine learning in recent years have relieved the difficulty of solving computer-hard problems to some degree, it still relies on large scale of training data which needs to be labeled manually. Crowdsourcing is a calculation paradigm to solve computer-hard tasks effectively and efficiently. It organizes large scale of human beings as workers through the Internet to solve problems. For decades, crowdsourcing has been applied in various areas. Many crowdsourcing platforms have achieved business success. A typical example is Amazon Mechanical Turk (AMT) , which is the biggest general-purpose crowdsourcing platform. According to a blog published by AMT , in a typical week there are millions of tasks that are processed by over a million workers all over the world. With the aid of the productive forces of AMT, the famous data set of image recognition, ImageNet , is labeled and has a huge impact. Besides, there are many people making a living on the income earned from AMT.

Despite the above achievements of crowdsourcing, the welfare of crowd workers is often neglected. The workers are usually regraded more as machines with some kinds of computing functions, which seems not a wise choice. On the one hand, although workers are viewed as machine, they are not as reliable or stable as computers. Thus, much effort has to be put to tame the uncertainty of workers by modeling their accuracy rates
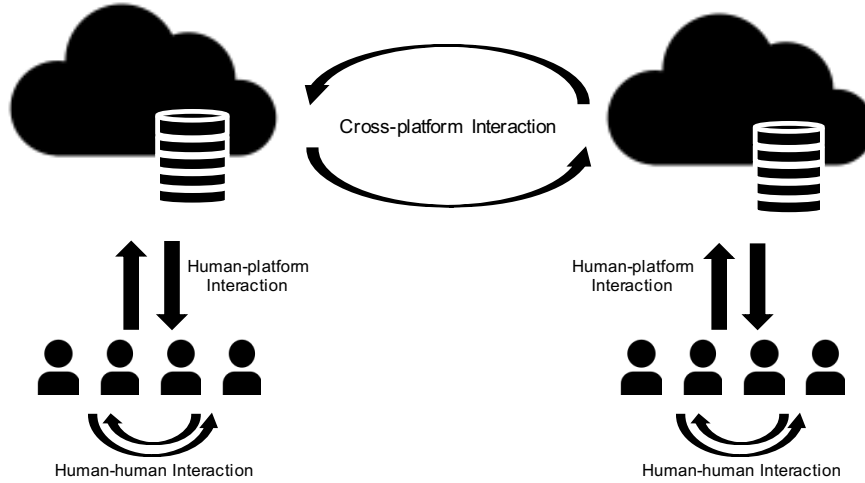
Figure 1: Categories of interactions.

and aggregating feedbacks. On the other hand, workers are more flexible and intelligent than machines. If their initiative can be stimulated, crowdsourcing platforms will be more productive and workers' welfare can also be enhanced.

To achieve the goals of stimulating the initiative of workers to further improve the effectiveness of crowd-sourcing platforms and helping the workers achieve self-actualization to make crowdsourcing a form of the future of work, this paper targets at the optimization of critical interactions in typical crowdsourcing ecosystems. To optimize the interaction between human workers and the platforms, *i.e.* the human-platform interaction, the *freewill* of human beings should be considered. The main part of human-platform interaction is task assignment, which assigns crowd tasks to suitable workers. Most existing studies on task assignment only consider how to maximize the utility and ignore whether the workers would like to perform the assigned tasks. To make task assignment more humanized, tasks should be assigned to workers based on the workers' will. More friendly methods of task assignment should be designed. To optimize the interaction between human workers, *i.e.* the human-human interaction, the *sociality* and *selfishness* of human beings should be considered. When the tasks are simple or well-paid, workers may complete them more efficiently. Thus, incentive mechanisms should be designed based on workers' rationality and selfishness. When the tasks are complex and hard to complete, the platforms need the workers to corporate with each other. Thus, the social relationships of workers should be considered too.

The rest of the paper is organized as follows. We categorize the interactions on crowdsourcing platforms in Sec. 2. We discuss how to optimize the human-platform interaction and human-human interaction in crowdsourcing in Sec. 3 and Sec. 4 respectively. We envision the future directions in Sec. 5 and conclude the paper in Sec. 6.

2   Types of Interactions in Crowdsourcing

An interaction process usually involves human workers and platforms as necessary participants. Therefore, we can divide interactions on crowdsourcing platforms into three categories: human-platform interaction, human-human interaction and cross-platform interaction, as shown in Figure 1.

Human-platform interaction. Interaction between human workers and platforms is an essential process for accomplishing crowdsourced tasks. Most human-platform interactive behaviors happen in the task assignment

stage, where the platform assigns tasks to appropriate human workers in order to achieve some specific targets such as maximizing the total profits. We have recognized that assigning tasks to human workers is rather different from assigning tasks to robots or AI agents, because human workers have free wills and they have rights to reject the assigned tasks or to choose whatever tasks they would like to carry out. Therefore, we will focus on the following two kinds of interactions during task assignment process. The first is called *task selection by workers*, where the platform displays all available tasks with their specifications, constraints and payments, and the workers can choose the expected tasks freely. This scenario is more common and can be found on many existing crowdsourcing platforms like Amazon Mechanical Turk (AMT) and Figure Eight. The second is called *task recommendation by platforms*, where the platform recommends tasks to workers actively and each worker can reject the assignment in which case the platform has to continuously recommend new tasks to them until the task is accepted. We will give a detailed introduction to them in Sec. 3.

Human-human interaction. Apart from the communication between human workers and platforms, there also exist interactions between different workers due to their social activities, which can influence the process of performing the tasks. For example, workers with close social relationships like friends or relatives can work together as a team to accomplish complicated tasks. Considering that human workers are rational and selfish, the platform cannot always expect them to stick together for the sake of finishing all the tasks more effectively and efficiently. Therefore, proper guidance of human-human interaction like incentive mechanisms can help improve the overall utility meanwhile respecting each individual's interests. In a way, we can encourage workers to form groups based on their social relations, common likes and dislikes. They will have *collaborative interaction* between each other, which can be more efficient than working alone. In another way, we can also allow friendly competitions between workers or groups and give the winners additional rewards. The *competitive interaction* can improve the efficiency of both sides and can also help the human workers find the value and fun in their efforts. We will introduce the above two human-human interactions in Sec. 4.

Cross-platform interaction. So far, we have discussed about the interactions inside the cycle of a single crowdsourcing platform and its workers. But we can also imagine a new type of interaction that exists between different platforms, which we call cross-platform interaction. Even though today's crowdsourcing platforms are isolated from each other, it is prospective that they cooperate with each other as a federation in the future to serve more tasks and give opportunities to more human workers. In such a scenario, the cross-platform interaction will play a rather important role. The challenges of designing proper cross-platform interaction methodology not only lie in dealing with heterogeneous systems, but also in taking care of the unique requirements of human workers: the *privacy concerns*. For example, suppose there is a supply constraint on Figure Eight and many tasks on the platform cannot be finished in time. Therefore, Figure Eight wants to federate with AMT which has sufficient worker supply. But workers on AMT do not always like to share their profiles and working histories with Figure Eight, which can influence the utility of assignment. The two platforms have to interact with each other to form a federation meanwhile no privacy leakage of workers will happen during the interaction process. We will discuss about such cross-platform interaction as future directions in Sec. 5.

3 Human-Platform Interaction: Task Assignment

In this section, we focus on the interaction management in task assignment between human workers and platforms. Specifically, we identify two types of human-platform interactions during the task assignment in crowdsourcing. In Section 3.1, we introduce the first type, *i.e. task selection by human workers*, where the workers can freely choose the desired tasks and a task is assigned to the first worker who selects it. In Section 3.2, we discuss the second type, *i.e. task recommendation by platforms*. In this type, the platform learns the preferences and profiles of the workers from the history records and then recommends the suitable tasks to the workers based on different methods. After that, the human workers can either accept the tasks or reject the tasks.
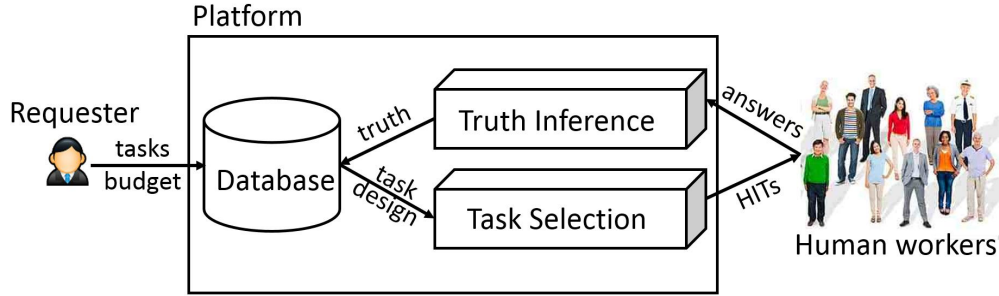
Figure 2: An illustration of the workflow in task selection [2].

3.1 Task Selection By Human Workers

In the process of task assignment, many crowdsourcing platforms apply the strategy of *task selection by human workers* in the interaction management, *e.g.* Amazon Mechanical Turk, Wikipedia, Quora, Yahoo!Answer. Specifically, these platforms usually first collect the tasks from the requesters, then decompose them into micro-tasks [1], and finally pack several micro-tasks into an HIT (*a.k.a.* task bin [1]). After the platforms release these HITs into well-designed user interface, human workers start to interact with the platforms in the process of task assignment. As shown in Figure 2, there are mainly two stages in this workflow: *task selection* and *truth inference*.

Task Selection. In this stage, human workers can freely choose the desired HITs and an HIT is assigned to the first worker who picks it. This mode is also known as *worker selected tasks (WST)* mode in the early studies [3, 4]. To attract the human workers, existing studies usually focus on *task design* in this type of interaction. Specifically, [2, 5] introduce *task types* and *task settings* as the fundamental problems in the task design. When designing the task types, the crowdsourcing platforms need to carefully choose the format of the tasks. For example, a problem with single choice is usually easier to be answer by the workers than the problem with multiple choices. As a result, more studies focus on the tasks with single choice (*e.g.* [6, 7, 8]) than the tasks with multiple choices (*e.g.* [9]). However, the tasks with multiple choices are relatively more profitable for the workers and more helpful for inferring the truth (*i.e.* the second stage). A good task design combines these two types in practice. For instance, Vasilis *et al.* [10] propose the approach called Waldo to detect the difficult tasks with multiple choices and only resolve them by single choice. When considering the task settings, the crowdsourcing platforms mainly determine the *pricing* and *timing* of the tasks. Usually, a high price can intuitively attract more workers, thereby reducing the total time to complete a task (*i.e.* low latency). However, a high price does not always indicate a good quality of the answer [11]. Therefore, a good task design needs to consider the trade-off between the pricing and timing. For instance, Yihan *et al.* [12] applies Markov Decision Process to determine the price and deadline of the tasks. Vasilis *et al.* [13] designs a dynamic-programming based method in the task setting while simultaneously choosing the proper task type.

Truth Inference. In this stage, the crowdsourcing platform will collect the answers from the workers after they accomplish their selected tasks. The challenge in this stage is how to infer the truth of each task based on the different tasks' answers collected from human workers [5]. Many solutions have been proposed to solve the problem of truth inference. A simple solution is to use majority voting, where the truth is viewed as the majority answer of the workers. However, this method overlooks the fact that the workers usually have different qualities in practice. To obtain the worker's quality, one way is to use the hidden test, where a golden question is injected in an HIT. Since the platform knows the truth of this golden question, it can estimate the workers' quality more accurately. However, the hidden test takes more money since the workers need to complete these extra tasks. Moreover, as reported in [14], the hidden tests may not improve much quality. Therefore, existing studies usually

focus on inferring the truth without the hidden tests. The most classic framework is Expectation Maximization (EM) . The EM method is firstly proposed in [15], which leverages all the collected answers from the workers, and updates each worker's quality and each task's true answer until convergence [2]. Other studies use the Graph-based methods [16, 17], where a worker's quality is represented as a node and is derived by the graphical model inference [18].

3.2 Task Recommendation By Platforms

There are many crowdsourcing platforms which take another mode *i.e.* task recommendation by platforms to interact with crowd workers during task assignment, such as Uber, Didi Chuxing [19, 20] and Facebook Editor. Early studies formulate the task recommendation problem as bipartite graph matching problem. Specifically, workers and tasks can be represented by the vertices in the bipartite graph and the score to evaluate the suitability of a worker and a task can be denoted by the weight of the edges. Then the problem is to obtain an optimal matching in the bipartite graph.

Statics Scenario. When the information of tasks and workers is known in advance (or before task recommendation), exact algorithms (*e.g.* Hungarian algorithm) can optimally solve the problem. Alternatively, Kazemi *et al.* [3] reduce the bipartite graph into an instance of the *maximum flow* problem and use the Ford-Fulkerson algorithm to obtain the exact result. They also consider some practical issues. For example, if for a task there are fewer workers who are suitable to perform it, the task should be recommended with higher priority. The idea of entropy can be borrowed to represent this priority. Another heuristic strategy is to iteratively assign the task to its most suitable workers. To reduce the computation cost of exact solutions, various greedy based methods are proposed. For example, [21] maximizes the total number of tasks which are accepted by the workers while considering a budget constraint.

Dynamical Scenario. In reality, the time when new tasks are submitted to the platform and the content of the tasks are unknown. Besides, the workers also log in the platforms dynamically. Thus, some studies address the task recommendation problem while considering the dynamics of tasks and workers. They formulate the problem as online bipartite graph matching. Karp *et al.* [22] first propose the RANKING algorithm which yields a competitive ratio $1 - 1/e$ under the *adversarial order model* and the ratio is proven to be the lower bound of any online algorithm. In [23], Tong *et al.* borrow the idea of secretary problem and devise a two-phase based framework. Their proposed algorithms achieve competitive ratios of 0.25 and 0.125 under the *random order model*. In [24], an comprehensive experimental study has been made and the greedy algorithm is proved to be competitive in real scenarios. More flexible and adaptive online matching algorithms have also been proposed in [25, 26, 27, 28]. In addition, Zhang *et al.* [29] focus on predicting the acceptance ratio of workers in task recommendation via machine learning techniques. They treat the rejected tasks by workers as new tasks that can be re-recommended to other workers. We refer readers to the comprehensive surveys [30] for more details on task assignment approaches.

4 Human-Human Interaction: Incentive Mechanism

Despite the essential interactions between humans and platforms, the sociality from humans has made human-human interactions more and more important in the future of work. These interactions, which we summarize as the incentive mechanisms, depict how workers interact with each other and focus on how to design mechanisms to motivate workers. To be more specific, on one hand, workers tend to be multi-skilled and social as crowdsourcing develops. Those positive interactions (*e.g.* complementary in skills or closeness in social contact) between humans can help improve the satisfaction of the workers and performance of the platforms. On the other hand, the freewill of humans enables selfish workers. Those malicious factors lead to inevitable competition, which

needs to be treated carefully. Correspondingly, we categorize the human-human interactions as collaborative interaction and competitive interaction. In the following of this section, we will discuss about them respectively.

4.1 Collaborative Interaction

An important kind of human-human interaction is the collaborative interaction. On one hand, tasks become more and more complicated and they usually require multiple skills to accomplish. Workers with different skills should collaborate to accomplish the complicated tasks. On the other hand, social factor may be helpful to improve the task quality and latency. For example, two workers which are good friends may accomplish the tasks with higher quality and lower latency. In a word, in this part we introduce two representative collaborative factors in human-human interaction, *i.e.* multi-skill factor and social factor.

Multi-skill Factor. Multi-skill factor is one of the significant factors to depict complicated tasks and has been widely studied in recent years. In early studies like [31, 32], the skills are formulated as the skill level value representing the probability that a worker submits the true answer. [33] utilizes the formulation of expertise level and applies a transfer learning model to estimate the expertise levels in all categories. To overcome the disadvantage that a single value cannot depict multiple skills, [34] extends the formulation to the environment of multiple skills and uses a vector called latent skill category to represent how a worker is familiar with different skills. [35] and [36] study the so-called team formation problem and propose the solutions with the help of social networks and road networks, respectively. In the future of work, despite of existing formulations like skill level and team formation, we may facet a more complicated multi-skill situation where the task requires a time dependency of skills. Under this situation, a task may require a worker with skill A and a worker with skill B sequentially, which requires a novel solution based on the accurate estimation of worker skills.

Social Factor. As human workers have their own social roles apart from workers on crowdsourcing platforms, the social factor can also influence the completion of tasks. Some work has used the social factors like relationships on social networks and communities [37] to help increase the crowdsourcing quality. In [38], the authors address the decision making question answering by crowdsourcing on micro-blog services. One of the most fundamental steps in addressing the problem is to estimate the rating score of different workers. The authors first construct a user-graph with Twitter data according to the workers' forwarding operation retweet, and then rank the users in the constructed graph. Therefore, expertises that are frequently followed by other twitters can be found through the user graph easily. In [39], the authors propose a novel community-based Bayesian label aggregation model, which can predict the labels of different workers by assuming that each worker belongs to a certain community. Some other work models the social relationships with game theory and reveals the importance of social factors in crowdsourcing. In [40], it considers the social team formation problem in crowdsourcing, where a team of socially connected professional workers can work together collaboratively. It focuses on designing truthful mechanisms according to workers' social structure, skill, and working cost to guarantee that a worker's utility is optimized when he behaves honestly. [41] directly studies the social factors in incentive mechanisms in crowdsourcing and finds that collaboration leads to better accuracy and more outputs. It reveals that in pair-work crowdsourcing, when one of the workers tries to exit the task, the partner is very likely to make him/her stay for higher payment and for enjoying the tasks together. Therefore, social furtherance incentives can create a win-win scenario for both the platform and the human workers.

Table 1 summarizes existing works on collaborative interactions. We summarize that both multi-skill and social factors raise attention but their formulation needs to be further completed to suit for more challenging situations.

4.2 Competitive Interaction

Despite the collaboration in human-human interaction, there also exists competitions in the interaction of crowdsourcing. Different from collaboration, malicious competitions may reduce the quality of the tasks or even

Table 1: Related works on collaborative interaction.

Reference	Interaction	Factor	Formulation
[31]	Collaborative	Multi-Skill Factor	Skill Level
[32]	Collaborative	Multi-Skill Factor	Skill Level
[33]	Collaborative	Multi-Skill Factor	Expertise Level
[34]	Collaborative	Multi-Skill Factor	Latent Skill Category
[35]	Collaborative	Multi-Skill Factor	Team Formulation
[36]	Collaborative	Multi-Skill Factor	Team Formulation
[38]	Collaborative	Social Factor	Rating Score
[39]	Collaborative	Social Factor	Label Prediction
[40]	Collaborative	Social Factor	Team Formulation
[41]	Collaborative	Social Factor	Social Furtherance

destroy the human-in-the-loop ecosystem. However, passive competitions can be beneficial if treated carefully. In this part we envision how to utilize competitions to encourage workers to participate in completing tasks with higher quality and lower latency. In details, depending on whether the reward is monetary or immaterial, we categorize the interaction as monetary factor and non-monetary factor.

Monetary Factor. Monetary is one of the most important factors that can influence how workers perform. Current studies usually formulate the monetary factor in competitive human-human interaction based on the game theory. In a word, a carefully designed pricing mechanism should be able to provide a well balance between payments (*e.g.* prices paid to workers) and to promote workers to submit the truthful information. Most works provide auction-based solutions to model the competitive interaction between humans. Workers need to bid their expected prices to the platform, and the platform chooses workers to accomplish tasks based on the proposed mechanisms. Notice that even there is no direct communication among workers, the mechanism which is known by all workers in fact provides an indirect competitive interaction among workers. Because of the selfishness of workers, the mechanisms should be truthful, *i.e.* the mechanism guarantees that workers cannot obtain more reward if they provide false information. Besides, another consideration of designing the mechanism is how to balance various constraints and objectives. For example, [42, 43, 44] aim at maximizing the utility (*i.e.* the value obtained from the workers) while making sure that the total budget is under a given threshold. Different from [42] and [43], [45] focuses on two different objectives: maximizing the number of accomplished tasks with constrained budget and minimizing total payment with least number of accomplished tasks. In future of work, the interaction may facet more challenging situations. Firstly, the private information may be more complicated (comparing with current research where each worker is associated with a private expected value). For example, each worker may privately possess the total order of preference to the tasks, the capability of different kinds of tasks, etc.. Besides, in future of work there can be other practical trade-offs and objectives like trade-off between payment and quality of tasks or between payment and latency of workers. These complicated private information and various objectives can raise more challenging problems in the competitive human-human interaction.

Non-Monetary Factor. Despite the dominant role of the monetary factor, none-monetary factor (*e.g.* gamification, volunteering etc.) also plays an important role in the competitive human-human interaction. Specifically, gamification can attract workers to join the procedure [46] and result in positive competitions that motivate workers to provide better services [47]. For example, [46] develops a series of games to attract workers to attain the crowdsourcing tasks. Further experiments show that the proposed games can improve the quality and latency of the accomplished tasks. Compared with [46], [47] combines the luck games (*e.g.* lottery) with crowdsourcing. Each worker is given one or more lotteries and which worker can be paid is related not only to the quality of the accomplished tasks, but also to the lottery results. In the future of work, non-monetary factor may become more and more important. An interesting study direction is to formally model how non-monetary factor like

Table 2: Related works on competitive interaction.

Reference	Interaction	Factor	Formulation
[45]	Competitive	Monetory Factor	Auction
[42]	Competitive	Monetory Factor	Auction
[43]	Competitive	Monetory Factor	Auction
[48]	Competitive	Monetory Factor	Collaboration
[46]	Competitive	Non-Monetory Factor	Relevance Assessments
[47]	Competitive	Non-Monetory Factor	Lottory

gamification produce positive competitive interaction and impact the performance of workers in crowdsourcing.

Table 2 summarizes existing works on competitive interactions. We summarize that most existing works focus on the interaction with monetary factor. Non-Monetary factor, although lacking in systematic study in crowdsourcing, can also play an important role in future work.

5 Future Direction: Cross-Platform Interaction

In the previous sections, we have discussed about human-platform and human-human interaction in existing crowdsourcing platforms. In this section, we are going to introduce a future direction: the cross-platform interaction. Nowadays, there is a large amount of commercial crowdsourcing platforms that divide the online job markets into many pieces. If these platforms can work together as a federation (as illustrated in Figure 3), task requesters will have a higher chance to meet more professional workers and workers will also have more career opportunities. However, sensitive information like worker profiles, professional skills and working histories cannot be shared freely between different platforms, for the purpose of preserving each human worker's privacy. Therefore, the cross-platform interaction should be more carefully designed and the major concern is how to preserve the privacy of human workers from different platforms.

There is rich literature on privacy-preserving schemes, including encryption methods like homomorphic encryption [49] and randomized algorithms like differential privacy [50]. Next, we will discuss about the possible shape of the interactive modules including cross-platform task assignment and cross-platform incentive mechanism, as well as the privacy-preserving techniques behind them.

Cross-platform task assignment. The interaction in cross-platform task assignment can be different from conventional task assignment where the platform has full knowledge of its tasks and workers. In a cross-platform task assignment scenario, workers will have opportunity to complete the tasks from other platforms. We conceive two types of interaction in cross-platform task assignment: interaction by recommendation and by negotiation. In the first type, surplus tasks on each platform can be shared with other platforms in the form of recommendation and the task specifications and the original links will be given. The recommended tasks from other platforms will be visible in the task pool of each platform, and the workers can select the preferred tasks freely. If they are interested in tasks from other platforms, they can have a quick registration through the link and become a cross-platform worker. Considering that there might be a huge number of pending tasks in the whole platform federation, one of the major challenge is to recommend for each worker a personalized ranking list of tasks so that they can locate their favorite tasks in a short time. A federated recommender system can be designed to protect the privacy of each worker while making personalized and accurate recommendations. Randomized algorithms like locality-sensitive hashing are possible solutions to protect private tags of workers in federated recommender systems. In the second type, the platforms will negotiate with each other at first. For example, there are surplus tasks on Figure Eight and the platform would like to borrow workers from AMT to finish these tasks. Figure Eight will negotiate with AMT, about the amount and types of the tasks and the profit shares meanwhile AMT can provide some information on workers to select appropriate tasks. Afterwards, the selected tasks will become
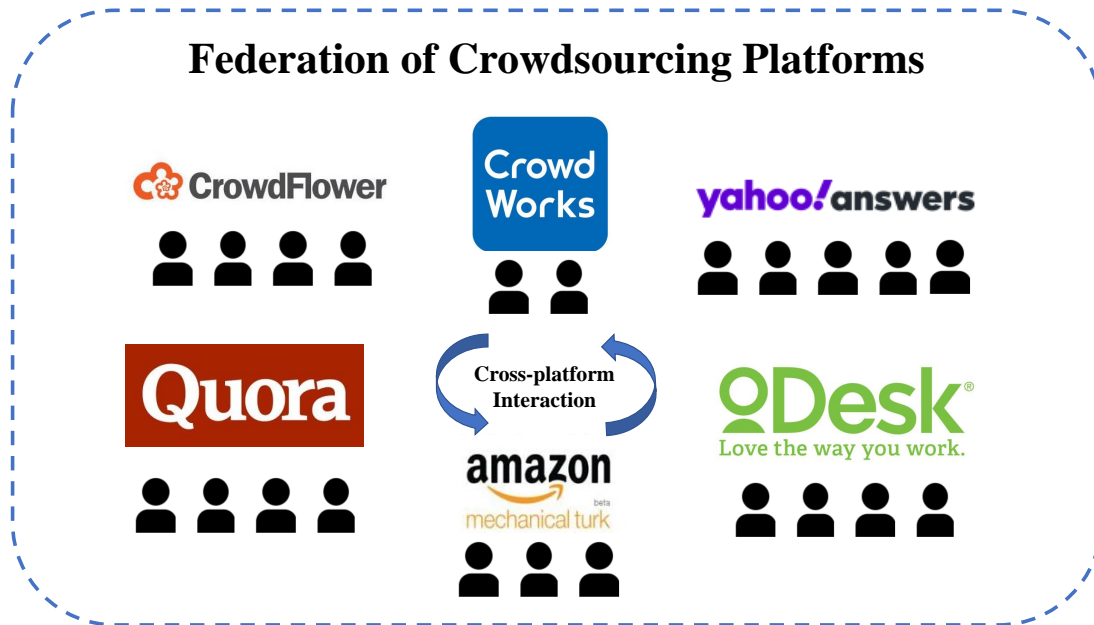
Figure 3: An illustration of the federation of crowdsourcing platforms

available to workers on AMT as new tasks. After the completion of tasks, AMT will return only the results and Figure Eight should know little about the specific workers who complete them. To protect the private profiles and working histories of the workers, there should be anonymization techniques during the negotiation. For example, AMT can only provide the summary or aggregated results of worker abilities for Figure Eight to decide which tasks can be shared. Another possible problem is that AMT can hide the information of profit shares from the workers if the payment by Figure Eight is much higher and how to ensure the trust and fairness remains challenging.

Cross-platform incentive mechanism. Similar to cross-platform task assignment, cross-platform incentive mechanism also faces privacy challenges. We mainly focus on incentive mechanisms in collaborative scenarios, where the privacy leakage is much more likely to happen. In a collaborative scenario, workers from different platforms can have a chance to form groups to perform cross-platform tasks. In one way, they can cooperate to perform a batch of simple tasks respectively and aggregate the results afterwards. For example, the federated platform publishes a task of labeling a large number of CT images. Expertises like doctors from different platforms can form a team to accomplish the tasks. They can divide the tasks according to their areas of expertise so the group work can be more efficient. In another way, each worker can also perform one or several steps in the working pipeline of complicated tasks. For example, the task is to tag all the objects in photos of street views and then to annotate the texts on street signs and buildings. It can be divided into several steps, including marking all the objects, annotating the texts and double-checking the results, where each step can be conducted by workers from different platforms. In both ways, the federated platform should design team formation algorithms based on social networks or ability tags of different workers, which also has risk of privacy leakage. Although there is plenty of work concerning privacy issues in social networks, how to preserve personal information while finding like-minded partners still remains a challenge.

6 Conclusion

Crowdsourcing has attracted much popularity in recent years, due to its high efficiency in dealing with computer-hard tasks by human workers. However, we have noticed the problem that humans are often treated as machines. To deal with such problem, managing the interaction among the workers and platforms is essential. In this paper, we categorize and analyze different interactions on crowdsourcing platforms, and discuss about the challenges and existing work on human-platform interaction and human-human interaction respectively. Finally, we envision a new type of interaction, the cross-platform interaction. We hope in the future of crowdsourcing, there will be a federation of platforms that offer much more opportunities to everyone everywhere and human workers will be treated as fully human with respect and dignity.

References

[1] Y. Tong, L. Chen, Z. Zhou, H. V. Jagadish, L. Shou, and W. Lv, "SLADE: A smart large-scale task decomposer in crowdsourcing," *IEEE TKDE*, vol. 30, no. 8, pp. 1588–1601, 2018.

[2] G. Li, J. Wang, Y. Zheng, and M. J. Franklin, "Crowdsourced data management: A survey," *IEEE TKDE*, vol. 28, no. 9, pp. 2296–2319, 2016.

[3] L. Kazemi and C. Shahabi, "Geocrowd: enabling query answering with spatial crowdsourcing," in *GIS*, 2012, pp. 189–198.

[4] D. Deng, C. Shahabi, and U. Demiryurek, "Maximizing the number of worker's self-selected tasks in spatial crowdsourcing," in *GIS*, 2013, pp. 314–323.

[5] G. Li, Y. Zheng, J. Fan, J. Wang, and R. Cheng, "Crowdsourced data management: Overview and challenges," in *SIGMOD*, 2017, pp. 1711–1716.

[6] D. Firmani, B. Saha, and D. Srivastava, "Online entity resolution using an oracle," *PVLDB*, vol. 9, no. 5, pp. 384–395, 2016.

[7] J. Wang, G. Li, T. Kraska, M. J. Franklin, and J. Feng, "Leveraging transitive relations for crowdsourced joins," in *SIGMOD*, 2013, pp. 229–240.

[8] Y. Zeng, Y. Tong, L. Chen, and Z. Zhou, "Latency-oriented task completion via spatial crowdsourcing," in *ICDE*, 2018, pp. 317–328.

[9] J. Wang, T. Kraska, M. J. Franklin, and J. Feng, "Crowder: Crowdsourcing entity resolution," *PVLDB*, vol. 5, no. 11, pp. 1483–1494, 2012.

[10] V. Verroios, H. Garcia-Molina, and Y. Papakonstantinou, "Waldo: An adaptive human interface for crowd entity resolution," in *SIGMOD*, 2017, pp. 1133–1148.

[11] S. Faradani, B. Hartmann, and P. G. Ipeirotis, "What's the right price? pricing tasks for finishing on time," in *HCOMP*, 2011.

[12] Y. Gao and A. G. Parameswaran, "Finish them!: Pricing algorithms for human computation," *PVLDB*, vol. 7, no. 14, pp. 1965–1976, 2014.

[13] V. Verroios, P. Lofgren, and H. Garcia-Molina, "tdp: An optimal-latency budget allocation strategy for crowdsourced MAXIMUM operations," in *SIGMOD*, 2015, pp. 1047–1062.

[14] Y. Zheng, G. Li, Y. Li, C. Shan, and R. Cheng, "Truth inference in crowdsourcing: Is the problem solved?" *PVLDB*, vol. 10, no. 5, pp. 541–552, 2017.

[15] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society*, vol. 39, no. 1, pp. 1–22, 1977.

[16] Z. Zhao, F. Wei, M. Zhou, W. Chen, and W. Ng, "Crowd-selection query processing in crowdsourcing databases: A task-driven approach." in *EDBT*, 2015, pp. 397–408.

[17] D. R. Karger, S. Oh, and D. Shah, "Iterative learning for reliable crowdsourcing systems," in *NIPS*, 2011, pp. 1953–1961.

[18] D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[19] Y. Tong, Y. Zeng, Z. Zhou, L. Chen, J. Ye, and K. Xu, "A unified approach to route planning for shared mobility," *PVLDB*, vol. 11, no. 11, pp. 1633–1646, 2018.

[20] Y. Zeng, Y. Tong, and L. Chen, "Last-mile delivery made practical: An efficient route planning framework with theoretical guarantees," *PVLDB*, vol. 13, no. 3, pp. 320–333, 2019.

[21] H. To, L. Fan, L. Tran, and C. Shahabi, "Real-time task assignment in hyperlocal spatial crowdsourcing under budget constraints," in *PerCom*, 2016, pp. 1–8.

[22] R. M. Karp, U. V. Vazirani, and V. V. Vazirani, "An optimal algorithm for on-line bipartite matching," in *STOC*, 1990, pp. 352–358.

[23] Y. Tong, J. She, B. Ding, L. Wang, and L. Chen, "Online mobile micro-task allocation in spatial crowdsourcing," in *ICDE*, 2016, pp. 49–60.

[24] Y. Tong, J. She, B. Ding, L. Chen, T. Wo, and K. Xu, "Online minimum matching in real-time spatial data: Experiments and analysis," *PVLDB*, vol. 9, no. 12, pp. 1053–1064, 2016.

[25] T. Song, Y. Tong, L. Wang, J. She, B. Yao, L. Chen, and K. Xu, "Trichromatic online matching in real-time spatial crowdsourcing," in *ICDE*, 2017, pp. 1009–1020.

[26] Y. Tong, L. Wang, Z. Zhou, B. Ding, L. Chen, J. Ye, and K. Xu, "Flexible online task assignment in real-time spatial data," *PVLDB*, vol. 10, no. 11, pp. 1334–1345, 2017.

[27] Q. Tao, Y. Zeng, Z. Zhou, Y. Tong, L. Chen, and K. Xu, "Multi-worker-aware task planning in real-time spatial crowdsourcing," in *DASFAA*, 2018, pp. 301–317.

[28] Y. Wang, Y. Tong, C. Long, P. Xu, K. Xu, and W. Lv, "Adaptive dynamic bipartite graph matching: A reinforcement learning approach," in *ICDE*, 2019, pp. 1478–1489.

[29] L. Zhang, T. Hu, Y. Min, G. Wu, J. Zhang, P. Feng, P. Gong, and J. Ye, "A taxi order dispatch model based on combinatorial optimization," in *KDD*, 2017, pp. 2151–2159.

[30] Y. Tong, Z. Zhou, Y. Zeng, L. Chen, and C. Shahabi, "Spatial crowdsourcing: a survey," *The VLDB Journal*, 2019. [Online]. Available: https://doi.org/10.1007/s00778-019-00568-7

[31] C. Ho and J. W. Vaughan, "Online task assignment in crowdsourcing markets," in *AAAI*, 2012.

[32] C. Ho, S. Jabbari, and J. W. Vaughan, "Adaptive task assignment for crowdsourced classification," in *ICML*, 2013, pp. 534–542.

[33] Z. Zhao, D. Yan, W. Ng, and S. Gao, "A transfer learning based framework of crowd-selection on twitter," in *KDD*, 2013, pp. 1514–1517.

[34] Z. Zhao, F. Wei, M. Zhou, W. Chen, and W. Ng, "Crowd-selection query processing in crowdsourcing databases: A task-driven approach," in *EDBT*, 2015, pp. 397–408.

[35] D. Gao, Y. Tong, J. She, T. Song, L. Chen, and K. Xu, "Top-k team recommendation and its variants in spatial crowdsourcing," *Data Science and Engineering*, vol. 2, no. 2, pp. 136–150, 2017.

[36] W. Wang, Z. He, P. Shi, W. Wu, and Y. Jiang, "Truthful team formation for crowdsourcing in social networks: (extended abstract)," in *AAMAS*, 2016, pp. 1327–1328.

[37] L. Zhang, T. Song, Y. Tong, Z. Zhou, D. Li, W. Ai, L. Zhang, G. Wu, Y. Liu, and J. Ye, "Recommendation-based team formation for on-demand taxi-calling platforms," in *CIKM*, 2019, pp. 59–68.

[38] C. C. Cao, J. She, Y. Tong, and L. Chen, "Whom to ask? jury selection for decision making tasks on micro-blog services," *PVLDB*, vol. 5, no. 11, pp. 1495–1506, 2012.

[39] M. Venanzi, J. Guiver, G. Kazai, P. Kohli, and M. Shokouhi, "Community-based bayesian aggregation models for crowdsourcing," in *WWW*, 2014, pp. 155–164.

[40] W. Wang, Z. He, P. Shi, W. Wu, Y. Jiang, B. An, Z. Hao, and B. Chen, "Strategic social team crowdsourcing: Forming a team of truthful workers for crowdsourcing in social networks," *IEEE TMC*, vol. 18, no. 6, pp. 1419–1432, 2019.

[41] O. Feyisetan and E. Simperl, "Social incentives in paid collaborative crowdsourcing," *ACM TIST*, vol. 8, no. 6, pp. 73:1–73:31, 2017.

[42] A. Singla and A. Krause, "Truthful incentives in crowdsourcing tasks using regret minimization mechanisms," in *WWW*, 2013, pp. 1167–1178.

[43] N. Anari, G. Goel, and A. Nikzad, "Mechanism design for crowdsourcing: An optimal 1-1/e competitive budget-feasible mechanism for large markets," in *FOCS*, 2014, pp. 266–275.

[44] Y. Tong, L. Wang, Z. Zhou, L. Chen, B. Du, and J. Ye, "Dynamic pricing in spatial crowdsourcing: A matching-based approach," in *SIGMOD*, 2018, pp. 773–788.

[45] Y. Singer and M. Mittal, "Pricing mechanisms for crowdsourcing markets," in *WWW*, 2013, pp. 1157–1166.

[46] C. Eickhoff, C. G. Harris, A. P. de Vries, and P. Srinivasan, "Quality through flow and immersion: gamifying crowdsourced relevance assessments," in *SIGIR*, 2012, pp. 871–880.

[47] M. Rokicki, S. Chelaru, S. Zerr, and S. Siersdorfer, "Competitive game designs for improving the cost effectiveness of crowdsourcing," in *CIKM*, 2014, pp. 1469–1478.

[48] S. K. Mridha and M. Bhattacharyya, "Introducing collaboration in competitive crowdsourcing markets," *IEEE Intelligent Systems*, vol. 34, no. 1, pp. 23–31, 2019.

[49] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *STOC*, 2009, pp. 169–178.

[50] C. Dwork, "Differential privacy," in *ICALP*, 2006, pp. 1–12.