Skill-and-Stress-Aware Assignment of Crowd-Worker Groups to Task Streams

Katsumi Kumai
University of Tsukuba
katsumi.kumai.2015b@mlab.info

Masaki Matsubara
University of Tsukuba
masaki@slis.tsukuba.ac.jp

Yuhki Shiraishi
Tsukuba University of Technology
yuhkis@a.tsukuba-tech.ac.jp

Daisuke Wakatsuki
Tsukuba University of Technology
waka@a.tsukuba-tech.ac.jp

Jianwei Zhang
Iwate University
zhang@iwate-u.ac.jp

Takeaki Shionome
Teikyo University
shionome@ics.teikyo-u.ac.jp

Hiroyuki Kitagawa
University of Tsukuba
kitagawa@cs.tsukuba.ac.jp

Atsuyuki Morishima
University of Tsukuba
mori@slis.tsukuba.ac.jp

Abstract

Worker-task assignments represent one of the critical issues in crowdsourcing, as they affect the quality of task results. This study addresses the problem of forming worker groups assigned to the same task in a task stream that requires more than one worker. We introduce a worker-group queue model that covers practical and common scenarios for task-stream crowdsourcing, and compare three strategies in terms of the skill balance among worker groups, the quality of the final outputs, the number of worker re-assignments of workers, and psychological stress felt by workers. We found that one of the compared strategies that employs multiple worker queues yields good results based on these measures.

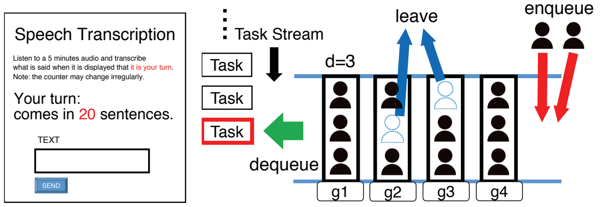

Speech Transcription
Listen to a 5 minutes audio and transcribe what is said when it is displayed that it is your turn.
Note: the counter may change irregularly.
Your turn:
comes in 20 sentences.
TEXT
SEND
Task Stream
Task
Task
Task
dequeue
d=3
leave
enqueue
g1
g2
g3
g4

Figure 1: (Left) Counter is displayed on the task screen of each worker, which counts down until the time that the worker should perform the task. (Right) Worker-group queue for task-stream crowdsourcing. Worker groups are restructured when workers join and leave.

1 Introduction

Worker-task assignments represent a critical issue in crowdsourcing because they affect the quality of task results. As most crowdsourcing requires more than one worker to perform the same task, addressing the problem in such cases is important. For example, the quality of the results computed by an aggregation method is considerably affected by the worker having the highest qualities among all workers performing the same task (Zhang et al. 2016) or the average skill of workers in a group (Lasecki et al. 2012).

This study addresses the problem of forming worker groups assigned to tasks in a task stream that require more than one worker. This setting appears in many scenarios. For example, if we want to crowdsource the task of producing captions for the video of a baseball game generated by relay broadcasting, we must generate a stream of tasks in which workers are asked to produce captions for a short video (Kacorri, Shinkawa, and Saito 2014; Deshpande et al. 2014).

In such a situation, we have to have workers wait for being assigned to tasks for a while for having tasks performed in a timely manner. It is well-known, however, that people feel stressed while waiting and that letting people know how long they have to wait is an important factor to alleviate their psychological stress (Nie 2000). Therefore, enabling workers to observe when their turn will arrive is crucial. For that purpose, a *counter* can be displayed on the task screen of each worker, which counts down the time until the worker should perform the task (Figure 1 (left)). With this counter, workers can prepare for their turn.

The situation can be naturally illustrated by the model in Figure 1 (right), which we call a *worker-group queue model*. In the model, workers who agree to perform tasks are added to worker groups in a worker-group queue. Unlike ordinary queues, it has different *units* for the enqueue and dequeue operations. We dequeue *groups* from the head of the queue, whereas we enqueue *workers* to the queue.

The worker-group queue model can also be applied to *retainer scenarios*, in which we retain workers who agree to perform tasks during a certain period of time (Bernstein et al. 2011). In this type of scenario, workers who have performed a task will be added to the worker-group queue repeatedly during a certain period. The worker-group queue model is often useful for volunteer-based crowdsourcing such as the task of having an audience produce captions for a lecture, because it avoids assigning only a few workers to many tasks.

The objective is that we want to balance the skills of workers in the worker groups. If the skills are balanced among groups, aggregations of the task results should be uniform in quality, which would avoid dissatisfaction caused by the curve of quality utility (Trestian et al. 2012). To make the skills balanced, she/he often needs to be inserted into groups other than the last one in the queue, according to her/his skill. In addition, in crowdsourcing, workers can join and leave a queue. Therefore, we must restructure worker groups in a queue based on worker behaviors.

On the other hand, this restructuring causes psychological stress in workers such as confusion or irritation. For example, if the counter suddenly jumps from 20 to 2, the worker might be surprised because she/he may not be fully prepared for the coming task. As we will demonstrate, a clear tradeoff exists between dynamically optimizing the distribution of skills among groups and the number of re-assignments of workers that often cause psychological stress.

This study presents three strategies that implement the aforementioned worker-group queue model. They are on different points of the trade-off line between the balance of skills among groups and stress caused by re-assignments of workers. Interestingly, we find that one of them achieves balanced skill groups with small changes of counters, thus causing low psychological stress in workers.

The contributions of this study are as follows:

Worker-group Queue Model and Skill-aware Worker Assignment. We present a model that covers many typical scenarios of task-stream crowdsourcing, with which we discuss how to assign workers to tasks to improve the quality of task results. The model naturally leads to consideration of the novel problem of re-assigning workers among task groups, triggered by workers when they join or leave.

Principled Strategies. We demonstrate three principled strategies and provide theoretical results. The first is an extreme strategy that minimizes the distances of worker movements. It does not consider the skills among groups. The second is another extreme case that always maintains the best-balanced skill groups. However, the problem is NP-complete and does not consider the distance of worker movements. The third lies between the two extreme cases and considers both factors to some extent. We show the results of the worst-case analysis.

Extensive Experiments. We compared the three strategies with both simulations and experiments with real-world crowd workers, in the skill distributions of task groups, data quality of the aggregated task results, the distances of worker movements, and the psychological stress to workers. We found that the third strategy achieves good skill balance with a set of low movements of workers and less psychological stress. The stress was observed in terms of both the NASA task load index (NASA-TLX) and word error rates.

Our key findings are as follows. First, there exists a worker re-assignment strategy that achieves both skill balance among worker groups and less re-assignment of workers. Second, workers' psychological stress is affected by re-assignments in worker-group queues. And finally, the effects of the re-assignment strategies on the task-result quality depend on workers' skills that are determined by the task and the set of workers.

The remainder of this paper is organized as follows. Section 2 reviews related studies. Section 3 describes the three strategies for worker-group queues. Section 4.1 compares the different strategies. Section 5 concludes the study.

2 Related Work

Crowdsourcing is currently expanding into many fields, including but not limited to character recognition (Simmons 2010), protein folding (Cooper et al. 2010), Web translation (von Ahn 2013), and image description (Bigham et al. 2010).

Real-time Crowdsourcing. Recently, real-time crowdsourcing has been attracting attention. VizWiz is considered the first nearly real-time crowdsourcing system (Bigham et al. 2010). Scribe is a system for real-time captioning by crowds, which uses a queuing model for a worker pool (Lasecki et al. 2012) (Naim et al. 2013). Bernstein et al. (2011) developed techniques that could be used to recruit synchronous crowds in two seconds and use them to execute complex search tasks in 10 seconds. Kacorri, Shinkawa, and Saito (2014) proposed a caption editing system that harvested crowd-sourced work for the task of video captioning using a game-like interface. Deshpande et al. (2014) developed a web-based crowdsourcing editor that corrected captions for video lectures. We note that most of those systems assume some type of worker queuing model. However, they do not consider worker skills in worker assignments to tasks.

Worker Property. In earlier systems, to achieve collaboration among multiple workers, workers must wait until a sufficient number of total workers are available (von Ahn and Dabbish 2005). Another research proposed producing event reports by using a combination of local and remote workers (Agapie, Teevan, and Monroy-Hernández 2015). Nushi et al. (2015) considered the diversity of workers to avoid crowdsourcing redundancy. We also solve the convergence of the same types of workers in groups, but our work focuses on worker skills rather than worker types, which can optimize the worker-group assignment even in the case when workers come from homogeneous sources.

Task Assignment. Various techniques for assigning tasks to workers have been developed (Ho and Vaughan 2012), (Ho, Jabbari, and Vaughan 2013), (Difallah, Demartini, and Cudré-Mauroux 2013), (Difallah et al. 2015). SmartCrowd is a framework for optimizing task assignments in knowledge-intensive crowdsourcing (Roy et al. 2015). This framework optimizes task assignments by forming optimal groups in advance. Kobren et al. (2015) presented techniques to dynamically assign tasks and present dynamic goals to workers. However, our work optimizes task assignments by restructuring groups dynamically. In contrast to group-restructuring methods for data items (Comer 1979) (Bayer and McCreight 1970), we must address workers' psychological stress generated by their re-assignment.

Data Quality. Data quality is a critical factor that has been analyzed extensively (Sheng, Provost, and Ipeirotis 2008), (Wang, Ipeirotis, and Provost 2017), (Karger, Oh, and Shah 2011), (Liu, Peng, and Ihler 2012), (Sarma, Parameswaran, and Widom 2016). In many cases, the quality of aggregation results depends on the skill distribution in a worker group.

For example, the highest skill (Zhang et al. 2016) and the average skill (Lasecki et al. 2012) are used. Our scheme works well in situations in which we wish task results to reflect a uniform level of quality. We also stress that our scheme is independent of a method for finding spam workers; once we identify them, we can easily implement a spam removal process by regarding them as workers who want to leave a group. Therefore, our method can be combined with other quality control methods that identify bad (i.e., spam) workers (Ipeirotis, Provost, and Wang 2010) (Le et al. 2010).

3 Worker-group Assignment

This section first provides definitions of the components of the worker-group queue model (Figure 1 (right)). In this study, we 1) perform a stream of tasks, to each of which we must assign d workers (d is the size of the worker group), and 2) have a set of workers who agree to perform the tasks. We then must formulate groups of workers and assign them to the tasks arriving from the task stream.

This section describes three strategies based on trade-off points between skill balance among groups of workers and the number of re-assignments of workers.

3.1 Worker Model

We assume that crowd workers have different skills and freely join and leave groups. In our model, the skill of each worker w is encoded by a numerical value that represents her/his skill. For example, the typing skill of a worker can be encoded by 1 minus the average word error rate (WER) of her/his typing results. Addressing the multiple types of skills of workers should be an interesting future study. Another interesting topic is how to deal with skill improvements. A possible approach is to constantly evaluate task results to measure and update skills, but this is also a future work.

3.2 Worker-group Queue

Let W be the set of workers who agreed to perform tasks. Given an integer d, a *worker-group queue* or wgq of worker groups is represented by a list $[g_1, g_2, \ldots g_{|wgq|}]$, where $W = g_1 \oplus g_2 \oplus \ldots \oplus g_{|wgq|}$ and $|g_i| = d$. Figure 1 (right) shows an example. Here, $|W| = 12$, $|wgq| = 4$ and $d = 3$.

We define three operations of wgq as follows:

- wgq`.dequeue()` returns g_1 from wgq and re-assigns the index of wgq so that g_1 always refers to the head of wgq. This operation is used to assign workers in g_1 to the task. The counters of the task screen are updated when the operation is performed.
- wgq`.enqueue(`w'`)` adds w' to a worker group in wgq.
- wgq`.leave(`w'`)` removes w' from wgq.

Note that unlike ordinary queues, the enqueue and dequeue operations are different in the *unit* of enqueued or dequeued items. We dequeue *groups* from the head of the queue, whereas we enqueue *workers* to the queue. In addition, we employ a special operation to remove workers who want to leave any group in the queue.

The worker-group queue model covers many typical scenarios in crowdsourcing in which:

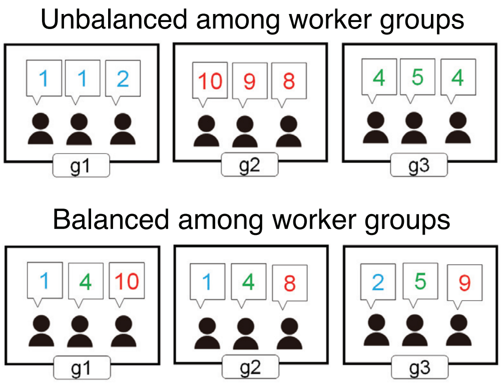


Figure 2: (Top) Unbalanced skills among groups. (Bottom) Balanced skills among groups. In the former, only low-skilled workers are present in g_1, which causes unsatisfactory results. On the other hand, in the latter, the averages and variances of skills in the groups are similar, which reduces unsatisfactory results.

- we have a stream of tasks,
- we have a worker pool that workers can freely join or leave, and
- each task should be performed by more than one worker.

The model can also deal with *worker retainer scenarios*, in which we *reuse* workers who performed a task. In the scenarios, workers in g_1 are enqueued again to the worker group queue after they perform their given tasks.

3.3 Problem Definition and Challenge

According to the curve of quality utility (Trestian et al. 2012), an unbalanced skill-group formulation as shown in Figure 2 (top), in which only low-skilled workers are present in g_1, causes unsatisfactory results. Instead, a balanced skills group formulation as shown in Figure 2 (bottom), in which the averages and variances of skills in the groups are similar, reduces unsatisfactory results.

Definition 1 (Best skill-balanced)**.** Worker groups are *best skill-balanced* if the following conditions hold:

- For every worker group, the i-th highest skilled worker in the group is guaranteed to be one of the j-th highest skilled workers among all workers where $i = \lceil j/d \rceil$, and
- The variance of the averages of worker skills in the groups is minimum in the assignments that satisfy the first condition. □

In short, each worker group has a similar distribution of worker skills and the variance of their averages is low. The assignments are suitable for many aggregation strategies such as those that depend on: top-skilled workers, skill average, and least-skilled workers, to produce task results of consistent quality.

Unfortunately, computing the best skill-balanced groups is not easy:

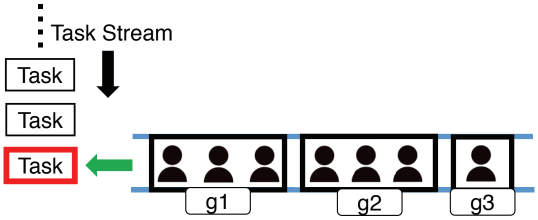


Figure 3: Single worker queue. This strategy produces fewer and shorter-distance re-assignments of workers. However, the strategy does not consider skill balance at all and is expected not to work well with respect to skill balance among worker groups.

Theorem 1 (Complexity)**.** Computing the best-balanced worker groups for a given set of workers is NP-complete.

Proof outline. The problem includes the three-partition problem as a subproblem. □

In addition, to maintain skill balance, workers often are re-assigned to other groups while they join and leave freely. However, workers may feel stressed if they are re-assigned frequently, because they mentally prepare for the tasks they expect to be assigned to.

Our challenge is to find solutions that maintain a good skill balance among worker groups while minimizing re-assignment of workers.

3.4 Strategies

This section describes three strategies for different points on the trade-off line between skill balance among groups and the number of worker re-assignments.

Single-worker Queue The single-worker queue (SWQ) implements the worker-group queue by using a single queue of workers (Figure 3). If a worker is located at the j-th in the worker queue, the group index number of the worker is defined as $\lceil j/d \rceil$. Each operation is implemented as follows:

- wgq.dequeue(): remove the first d workers from the worker queue.
- wgq.enqueue(w'): add worker w' to the end of the queue.
- wgq.leave(w'): remove worker w' from the worker queue.

SWQ produces fewer and shorter-distance re-assignments of workers. However, it does not consider skill balance at all.

Best Skill-balanced The best skill-balanced (BSB) strategy directly manages the balance of skills among groups in a worker-group queue; every time a worker joins or leaves, the strategy re-assigns workers to the worker groups in the queue so that the skill distribution among groups remains balanced. Each operation is implemented as follows:

- wgq.dequeue(): remove the first group (as well as the workers in that group) from the worker-group queue.

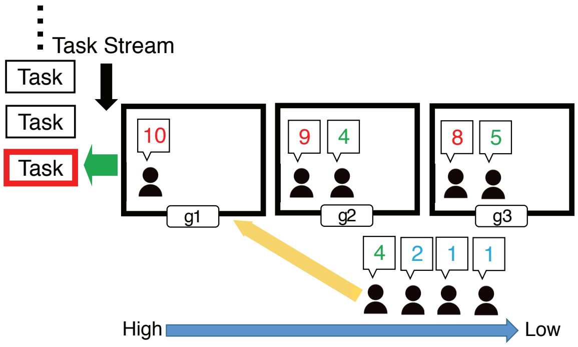


Figure 4: Best skill-balanced strategy. Each worker group has a similar distribution of worker skills and the variance of their averages is low. However, every time a worker joins or leaves, re-assignments occur.

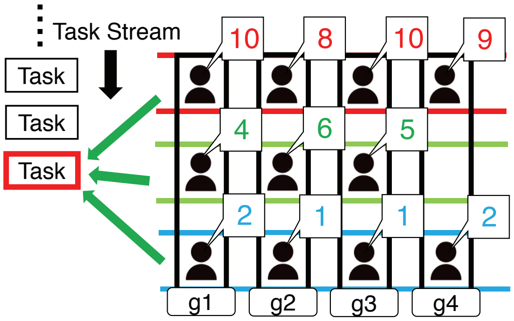


Figure 5: Skill-layered worker queue to reduce re-assignments while striving to maintain skill-balanced groups. The re-assignment distance is minimized so that workers do not feel stressed.

- wgq.enqueue(w'): re-assign all workers in the queue and w' to the groups in the queue. Because we have additional worker w', the number of groups in the queue may increase. Let $|W|$ be the number of workers in the worker-group queue before adding w' to it. The number of resulting groups will be $\lceil (|W| + 1)/d \rceil$.
- wgq.leave(w'): remove w' from the worker queue and re-assign all workers in the queue (without w') to the groups in the queue.

Skill-layered Worker Queue The skill-layered worker queue (SLWQ) implements the worker-group queue using d queues of workers (Figure 5). The underlying idea is to reduce the number of re-assignments, while striving to maintain skill-balanced groups. The distance of re-assignment is minimized so that workers do not feel stressed as shown in prior research (Kumai et al. 2017).

The queues are associated with different skill levels, and when a worker is enqueued to a (virtual) worker-group queue, his or her skill value is used to determine the worker queue that will accept him or her. Assume that a worker's skill value is the k-th highest among all workers in the

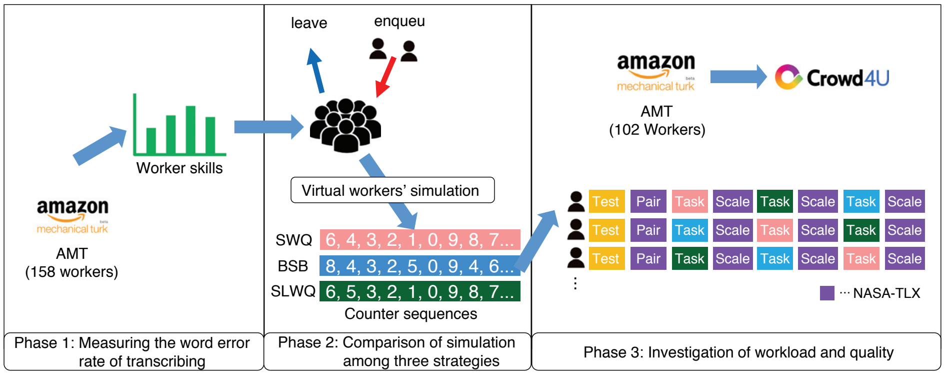

leave
enqueu
Worker skills
amazon mechanical turk
AMT
(158 workers)
Virtual workers' simulation
SWQ
6, 4, 3, 2, 1, 0, 9, 8, 7...
BSB
8, 4, 3, 2, 5, 0, 9, 4, 6...
SLWQ
6, 5, 3, 2, 1, 0, 9, 8, 7...
Counter sequences
amazon mechanical turk
Crowd4U
AMT
(102 Workers)
Test Pair Task Scale Task Scale Task Scale
Test Pair Task Scale Task Scale Task Scale
Test Pair Task Scale Task Scale Task Scale
⋮
··· NASA-TLX
Phase 1: Measuring the word error rate of transcribing
Phase 2: Comparison of simulation among three strategies
Phase 3: Investigation of workload and quality

Figure 6: Overview of experiments.

queue at that time. The first queue accepts the worker if $k \leq \lceil |W|/d \rceil$, and the j-th ($j \geq 2$) queue holds workers when $\lceil (j-1)|W|/d \rceil < k \leq \lceil j|W|/d \rceil$, and so on. The worker group g_l is defined as the set of l-th workers of the d worker queues. Unlike the BSB strategy, workers move only within the current queue and do not move from one queue to another. Each operation is implemented as follows.

- wgq.dequeue(): return the first workers of the d queues and remove those workers from the queues they were in.
- wgq.enqueue(w'): add w' to the end of one of the worker queues according to its skill value.
- wgq.leave(w'): remove w' from the worker queue.

If a queue becomes empty, SLWQ adopts the same strategy as BSB at that time. We assume, however, that SLWQ is used in the case where we always have sufficient workers to maintain queues.

SLWQ has low computational complexity but does not necessarily output the BSB groups. The following theorem shows the worst-case outputs for the case where workers are chosen randomly from the worker set.

Theorem 2 (Worst Case). Let min and max be the lowest and highest skill values of workers in a worker group queue with the group size being d. Let ave be the average of worker skills of the best-skill-balanced groups, while ave' is the one generated by SLWQ, with the given worker-group queue. Then, the following holds:

$$|ave' - ave| < \frac{max - min}{d} \tag{1}$$

Proof outline. A worst case happens when the ave is the largest possible one and ave' is the smallest possible one. In that case, the value of the i-th worker queue for ave' is the largest value of the $i-1$ worker queue. □

This indicates that the worst case depends on the minimum and maximum skill values, and on the size of groups. Note that the assumption may not necessarily hold in practice. In the following section, we will show that SLWQ usually outputs fairly good groups.

4 Experiments

We compared the three assignment strategies described in Section 3.4 regarding skill balance, the number of worker reassignments, and the final data quality. As shown later, we find that the SLWQ is superior to the others in that it obtains better results when we can correctly measure worker skills, and if not, it does not produce worse results.

4.1 Procedures

The experimental procedure consists of three phases as shown in Figure 6.

In Phase 1, we submitted the tasks to Amazon Mechanical Turk to measure the word error rates (WERs) of workers in transcribing the five audio clips. We obtained WERs of 158 workers for the five transcription tasks (5*158=790 in total).

In Phase 2, we conducted a simulation to compare the three strategies. For the simulation, we assumed that we had the 158 workers with their worker skills computed in two ways. Worker arrivals were determined by a Poisson process and staying times of workers were modeled by an exponential distribution. We also assumed that we had streams of one of the five transcription tasks.

In Phase 3, we evaluated the psychological stress of 102 workers when they transcribed the audio according to the counter used in the simulation of Phase 2. We measured the mental workload of the worker with NASA-TLX(Hart and Staveland 1988). The workers were recruited via Amazon Mechanical Turk and asked to perform the task generated by Crowd4U[1].

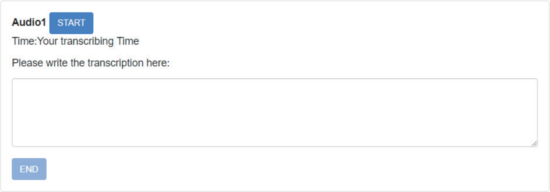
4.2 Settings

Tasks. We generated five audio transcription tasks, each of which asks a worker to transcribe a six-second audio clip taken from VOA news (Voice of America 1942). Table 1 shows the scripts of the five audio clips. We assume that each task is performed by more than one worker, and we use the A* search-based multiple sequence alignment strategy (Naim et al. 2013) to compute the final integrated result for each task.

[1] https://crowd4u.org

Table 1: Scripts excerpted from VOA News.

ID	Scripts of audio
1	Danleys lawyer told reporters she was in the Philippines to visit family paddock sent her a hundred
2	Emergency workers the senate intelligence committee revealed it is continuing to investigate
3	To collusion but were developing a clearer picture of what happened despite dropping no bomb
4	President Trump has reaffirmed his total confidence in secretary of state Rex Tillerson
5	But the places I come from we don't deal with that kind of petty nonsense and it is intended to do nothing but divide



Time:Your transcribing Time

Please write the transcription here:

Figure 7: Screenshot of transcription task. An audio clip excerpted from VOA News was played when the start button was pushed, and workers were asked to transcribe the audio.

Worker Skill. For each worker, we measured his/her WER (Wang, Acero, and Chelba 2003) for the tasks above, as it is a common evaluation index for transcription. Then, we used the WER to compute worker skill in two different ways.

4.3 Phase 1: Measuring the Word Error Rate of Transcribing

A total of 158 crowd workers were recruited to perform transcription tasks using Amazon Mechanical Turk with $ 0.3 reward per assignment. We didn't set any qualification for recruiting. Figure 7 is a screenshot of the task performed using the system.

Result 1. Figure 8 shows the WER distribution for each script. The horizontal axis represents WER and the vertical axis represents the number of workers who performed the transcription with the corresponding WER. Given a task (with a script i) and a worker j, $WER_{i,j}$ is calculated by the total number of substitutions (S), deletions (D), and insertions (I) divided by the number of words (N), as given in the following equation.

$$WER_{i,j} = (S + D + I)/N \tag{2}$$

The captions of transcribing tasks were made lower case and special characters were removed. The respective WER was calculated from those results. As shown in the figure, the scripts offer a variety of difficulty levels.

4.4 Phase 2: Comparison of Simulation among Three Strategies

We conducted a simulation to compare the three strategies. For the simulation, we adopted a retainer scenario in which worker arrivals were randomly determined by a Poisson process and staying times of workers had an exponential distribution. The workers in a dequeued group were assigned to a transcription task in the task stream. We used five task streams, where tasks in the same stream involved the same audio clip (one of five).

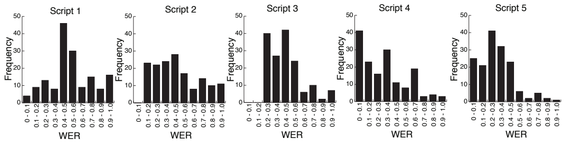


Figure 8: WER distribution of each script.

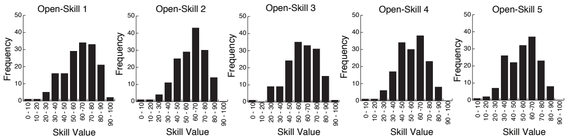


Figure 9: Distribution of open skill values. Open-Skill i shows the skill distribution estimated from the skills we measured on scripts ks s.t. $k \neq i$.

The strategies were evaluated based on the same factors of worker skill, worker arrival, worker staying times, and task assignment interval. We compared the three strategies regarding number of re-assignments, the distance of re-assignments, the average/variance skill of the workers assigned to the task, and the WER of the integrated final results generated by merging the captions produced by d workers.

Worker Skills Given a task with script i and a worker j, we computed the skill value of worker j for script i in the following manner:

$$OpenSkillValue_{i,j} = \frac{1}{N-1} \sum_{k \in \{1,\ldots,N\} \& \; k \neq i} (1 - WER_{k,j}) \times 100 \tag{3}$$

where N is the number of scripts (in this experiment $N = 5$). Although the open skill value does not necessarily represent the skill of each worker precisely, this worker skill type is used to predict the practical skills of workers based on their test results.

Figure 9 shows the distributions of open skill values. The horizontal axis represents the range of skill values and the vertical axis represents the numbers of workers who have the corresponding skill values. The open skill value was distributed in a form that approximated the normal distribution. The closer the distribution of worker skills was to the uniform distribution, the greater the expected differences in worker skills.

Other Settings Details of the experiment are as follows:

Task streams. We generated five task streams, each of which contains 1000 tasks for the same script (one of the five scripts). The task arrives at a predetermined interval (6 s).

Size of each worker group. We set d, the size of a worker group, to 4. Each task was performed by four workers.

Worker arrivals and staying times of workers. We assumed that worker arrivals were determined by a Poisson process and staying times of workers had an exponential distribution. Arrivals occurred at rate $\lambda = 10$ according to a Poisson process. Staying times of workers had an exponential distribution with rate parameter $\mu = 5$.

Integration of partial captions Integrating worker results can be thought of as multiple sequence alignment (MSA). In a previous study, an A* search-based MSA algorithm was developed (Naim et al. 2013). We apply the algorithm to the results of d workers to obtain each final integrated result.

4.5 Results of Phase 2

We evaluate simulation results based on the number of re-assignments, the distance of re-assignments, the average/variance skill of the workers assigned to the task, and the WER in a sentence, generated by merging the captions produced by d workers. Because there is no large difference in the distribution of open skills in Figure 9, we only show the results of script 1.

Result 2-I: Frequency and distance of re-assignments. Figure 10 shows histogram of the distance of re-assignments during 1000 tasks in a stream for each strategy. The number of re-assignments of the SLWQ and the SWQ were lower than that of the BSB. In the SWQ and the SLWQ, when a worker leaves, an irregular counter change occurs for the worker waiting behind the worker who left a group queue. The SLWQ can reduce the number of re-assignments of workers to a greater extent than the SWQ by using multiple queues. In contrast, to maintain the average and variance of skills among worker groups in the BSB strategy, workers are often re-assigned.

Here, the distance of re-assignments is expressed as a deviation between "the counter change that a worker assumed" and "the counter change that actually occurred". For example, in the case of a change from "5 more times" to "4 more times", the type of change is 0 because there is no difference between "the counter change that a worker assumed" and "the counter change that actually occurred". However, if the counter changes from "5 more times" to "2 more times," the counter change expected by the worker would be "4 more times.", so the actual change to "2 more times" represents a deviation. The distance of re-assignments at this time is -2.

The distance of re-assignments in the SWQ and SLWQ is only -1 or -2. Therefore, even if re-assignments occur, the stress in workers is expected to be low in the SWQ and SLWQ strategy. However, distance of re-assignments ranged from -16 to 16 in the BSB strategy. This would cause stress in workers.

Result 2-II: Average and SD of skills among worker groups assigned to the task

The averages of mean open skill among worker groups for all strategies were not very different. However, variation of SWQ was larger than that of the BSB and SLWQ since the SWQ strategy did not consider skill balance at all. Figure 11 shows the mean open skill values among worker groups assigned to task in script 1. For the clarity, we do not show all results for 1000 tasks, just the first 100 tasks. The mean open skill averages of SWQ, BSB, and SLWQ were 61.14, 62.46, and 61.55, while those standard deviation were 9.14, 2.78, and 3.65, respectively.

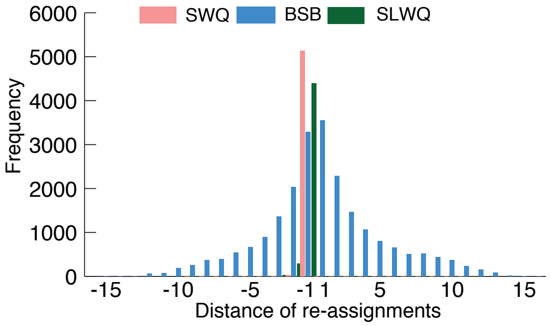


Figure 10: Number and distance of re-assignments in simulation of script 1.

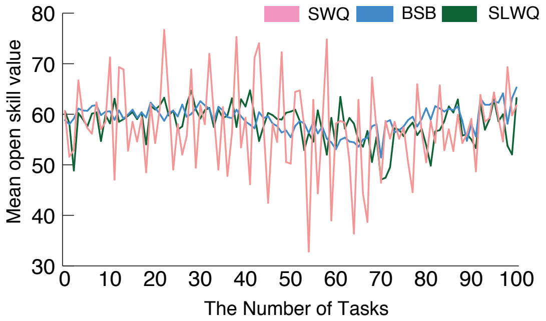


Figure 11: Mean open skill values among worker groups assigned to the task in script 1.

Result 2-III: WERs of the integrated results (open skill values). To evaluate the quality of transcription, an integration algorithm was applied to the task results of Phase 1 on the workers assigned to each task. Multiple comparison tests with Bonferroni correction were performed for statistical analysis. Figure 12 shows the box-and-whisker plots of the WER of the integrated results, which are generated by merging the results of d workers for each task. In script 1 and script 3, no significant difference in the average WER for 1000 tasks was observed. However, for script 2, script 4, and script 5, there were significant differences in each method. The result of script 4 showed that the average value of WER of the SWQ strategy was the worst. This suggested that task assignments performed while considering worker skills, improved the quality of the task results.

It can be seen that the SD of the SWQ was the worst, and variations occurred in the results of each task. The BSB and SLWQ exhibited low variance for each task. This suggested that by assigning tasks while considering worker skills, preventing large variations in the quality of each task result was possible.

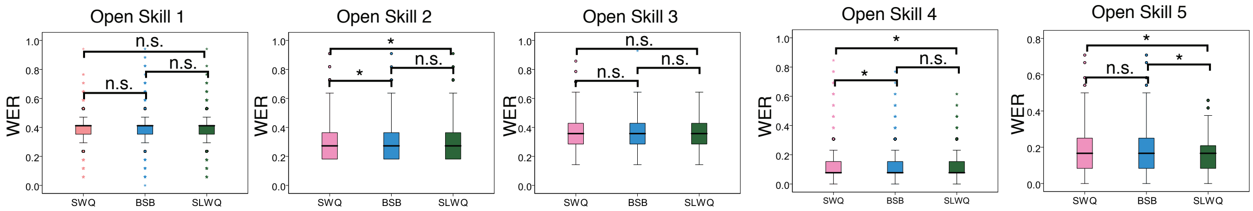


Figure 12: Box-and-whisker plots of WERs among 1000 integrated task results with open skill values.

4.6 Phase 3: Investigation of Workload and Quality

Phase 3 evaluates the stress of the workers when workers transcribe the audio according to the counter. From the simulation of Phase 2, how the counters of workers changed with each method was observed. We asked the workers to transcribe audio with this counter sequence, and then measured the mental workload of the workers with NASA-TLX (Hart and Staveland 1988).

NASA-TLX NASA-TLX is a subjective workload evaluation method used in many studies. It consists of six evaluation scales; mental demand, physical demand, temporal demand, work performance, effort, and frustration. Workers perform the tasks in two steps to measure the workload.

Pairwise Comparisons All pairs of the six evaluation scales are paired and displayed on the screen. Workers select the evaluation scale that represents the more important contributor to workload. A total of 15 ($_6C_2$) comparisons are made, covering all patterns, and the number of times the evaluation scale is selected is taken as the weight of each scale.

Scale Questions Workers evaluate task experience in each scale. Each scale is assigned a size of 0 to 100 and recorded.

Weighted Workload (WWL) Weighted Workload is a scaled evaluation point multiplied by the weight of the scale.

Tasks Workers were asked to listen to the two minutes audio, follow the counter, and transcribe the audio only when "Your Turn" is displayed. The starting position of the counter and the number of processing the task are individually determined by the counter sequence generated in Phase 2. The experiment was conducted through Amazon Mechanical Turk and Crowd4U, with a total of 102 results. For audio, we cut out two minutes from a five minutes news segment of "VOA News". Figure 7 shows the task screen. When the start button is pushed, the audio begins to play, and the counter starts to change. The worker performs the task only in her/his assigned section. In order to evaluate the WER, we adjusted the audio of the section where the worker finally performed the task to be the same for each worker.

Experimental Procedures Each worker evaluated the stress of three strategies in counter balanced order. Experiments were conducted using the following procedures.

- Test task (2 minutes). The workers transcribed the audio according to the counter sequence with no irregular change.
- Pairwise Comparisons. Based on the experience of the test task, workers selected the evaluation scale value that represented the more important contributor to workload.
- Transcribing tasks and scale questions (2 minutes + NASA-TLX scale questions, repeated 3 times). The workers transcribed the audio according to the counter sequence generated by each strategy in Phase 2. After task completion, workers evaluate the size of each scale based on the task experience.

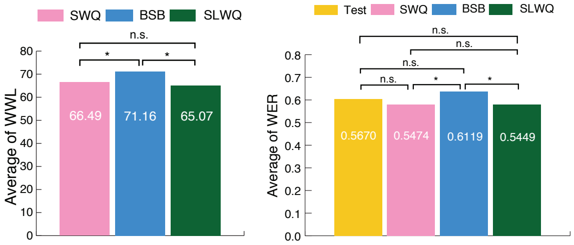


Figure 13: Average of Worker's WWL and WER

Result 3-I: WWL Figure 13 (left) shows the average worker's WWL. The Wilcoxon Signed rank test showed a significant differences between SWQ and BSB strategies, and between SLWQ and BSB strategies ($p < 0.05$).

This result suggests the BSB workload exhibits a larger difference than the other methods and thus the BSB increases worker stress because of irregular counter changes.

Result 3-II WER Figure 13 (right) shows the average of worker's WER. We evaluated the WER of the section where the worker transcribes the audio for the final time. The Wilcoxon test showed a significant differences between SWQ and BSB strategies, and between SLWQ and BSB strategies ($p < 0.05$). This is because the irregular change in the BSB strategy is larger than the other methods; it is difficult to grasp the timing at which you should transcribe the audio, so that the quality of the task results in work decline.

4.7 Discussions

Phase 1 results suggested that a worker's skill will be affected by the differences in difficulty among problems. It is not effective to measure workers' skills from the average of multiple results, because the open skill results were not specific enough. It is important not only to estimate workers' skills but also to estimate how well workers can achieve good results for each problem.

The experimental results of Phase 2 and 3 we obtained suggest that the SLWQ strategy can output task results with acceptable quality while minimizing stress. Interestingly, although BSB strategy was the best in phase 2, if the worker actually performs the task with counter sequence in Phase 3, individual worker results of BSB was worse than those of other strategies. As a result, although BSB can improve quality unless human stress is taken into consideration, in fact the results suggested that irregular changes make workers stressed and make the results worse. Note that SLWQ is better than SWQ in terms of the variance of WER, although their averages are comparable to each other.

Our experiment was not conducted under the best conditions. Specifically, the conditions were not ideal for BSB and SLWQ. First, the skill values we obtained may not have revealed the best skills for the tasks conducted in the experiment. This is because the tasks actually performed were different from those we used to measure skills. Second, the distribution of skills was a normal distribution and the skills of many workers were similar and approximated the average skill. These two factors mean that the skill-aware methods used were possibly less effective than the ideal condition.

Even under such conditions, SLWQ showed better performance in that it provided acceptable data quality with small stress and can be expected to perform well in many cases in terms of data quality and small stress.

5 Conclusion

This study proposed the worker group queue as a task assignment method for performing tasks that are continuously generated. We assumed that workers could join or leave a worker group freely. By devising a multiple-queue strategy for assigning tasks, we found that the psychological stress of workers could be reduced and the quality of each task could be maintained at a consistent level.

In this study, we assumed that all workers feel the same irrespective of their skill level in the simulation. However, the rate of departure and stress level would vary according to the skill set of the worker. We shall investigate worker-centric factors (Amer-Yahia and Roy 2016) in future work.

Acknowledgments

This work was partially supported by JSPS KAKENHI, Grant Number JP15K01056, JP16K16460, JP17K20022, Promotional Projects for Advanced Education and Research in NTUT, and JST CREST Grant Number JPMJCR16E3, Japan.

References

Agapie, E.; Teevan, J.; and Monroy-Hernández, A. 2015. Crowdsourcing in the field: A case study using local crowds for event reporting. In Gerber, E., and Ipeirotis, P., eds., *Proceedings of the Third AAAI Conference on Human Computation and Crowdsourcing, HCOMP 2015, November 8-11, 2015, San Diego, California.*, 2–11. AAAI Press.

Amer-Yahia, S., and Roy, S. B. 2016. Toward worker-centric crowdsourcing. *IEEE Data Eng. Bull.* 39(4):3–13.

Bayer, R., and McCreight, E. M. 1970. Organization and maintenance of large ordered indexes. In *Record of the 1970 ACM SIGFIDET Workshop on Data Description and Access, November 15-16, 1970, Rice University, Houston, Texas, USA (Second Edition with an Appendix)*, 107–141. ACM.

Bernstein, M. S.; Brandt, J.; Miller, R. C.; and Karger, D. R. 2011. Crowds in two seconds: enabling realtime crowd-powered interfaces. In Pierce, J. S.; Agrawala, M.; and Klemmer, S. R., eds., *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology, Santa Barbara, CA, USA, October 16-19, 2011*, 33–42. ACM.

Bigham, J. P.; Jayant, C.; Ji, H.; Little, G.; Miller, A.; Miller, R. C.; Miller, R.; Tatarowicz, A.; White, B.; White, S.; and Yeh, T. 2010. Vizwiz: nearly real-time answers to visual questions. In Perlin, K.; Czerwinski, M.; and Miller, R., eds., *Proceedings of the 23rd Annual ACM Symposium on User Interface Software and Technology, New York, NY, USA, October 3-6, 2010*, 333–342. ACM.

Comer, D. 1979. The ubiquitous b-tree. *ACM Comput. Surv.* 11(2):121–137.

Cooper, S.; Treuille, A.; Barbero, J.; Leaver-Fay, A.; Tuite, K.; Khatib, F.; Snyder, A. C.; Beenen, M.; Salesin, D.; Baker, D.; and Popovic, Z. 2010. The challenge of designing scientific discovery games. In Horswill, I., and Pisan, Y., eds., *International Conference on the Foundations of Digital Games, FDG '10, Pacific Grove, CA, USA, June 19-21, 2010*, 40–47. ACM.

Deshpande, R.; Tuna, T.; Subhlok, J.; and Barker, L. 2014. A crowdsourcing caption editor for educational videos. In *IEEE Frontiers in Education Conference, FIE 2014, Proceedings, Madrid, Spain, October 22-25, 2014*, 1–8.

Difallah, D. E.; Catasta, M.; Demartini, G.; Ipeirotis, P. G.; and Cudré-Mauroux, P. 2015. The dynamics of micro-task crowdsourcing: The case of amazon mturk. In *Proceedings of the 24th International Conference on World Wide Web, WWW 2015, Florence, Italy, May 18-22, 2015*, 238–247.

Difallah, D. E.; Demartini, G.; and Cudré-Mauroux, P. 2013. Pick-a-crowd: tell me what you like, and i'll tell you what to do. In *22nd International World Wide Web Conference, WWW '13, Rio de Janeiro, Brazil, May 13-17, 2013*, 367–374.

Hart, S. G., and Staveland, L. E. 1988. Development of nasa-tlx (task load index): Results of empirical and theoretical research. *Advances in psychology* 52:139–183.

Ho, C., and Vaughan, J. W. 2012. Online task assignment in crowdsourcing markets. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, July 22-26, 2012, Toronto, Ontario, Canada.*

Ho, C.; Jabbari, S.; and Vaughan, J. W. 2013. Adaptive task assignment for crowdsourced classification. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, 534–542.

Ipeirotis, P. G.; Provost, F.; and Wang, J. 2010. Quality management on amazon mechanical turk. In *Proceedings of the ACM SIGKDD Workshop on Human Computation*, 64–67.

Kacorri, H.; Shinkawa, K.; and Saito, S. 2014. Introducing game elements in crowdsourced video captioning by non-experts. In *International Web for All Conference, W4A '14, Seoul, Republic of Korea, April 7-9, 2014*, 29:1–29:4.

Karger, D. R.; Oh, S.; and Shah, D. 2011. Iterative learning for reliable crowdsourcing systems. In *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain.*, 1953–1961.

Kobren, A.; Tan, C. H.; Ipeirotis, P. G.; and Gabrilovich, E. 2015. Getting more for less: Optimized crowdsourcing with dynamic tasks and goals. In *Proceedings of the 24th International Conference on World Wide Web, WWW 2015, Florence, Italy, May 18-22, 2015*, 592–602.

Kumai, K.; Zhang, J.; Shiraishi, Y.; Wakatsuki, D.; Kitagawa, H.; and Morishima, A. 2017. Group rotation management in real-time crowdsourcing. In *Proceedings of the 18th International Conference on Information Integration and Web-based Applications and Services, iiWAS 2017, Salzburg, Austria, December 4-6, 2017*, 23–31.

Lasecki, W. S.; Miller, C. D.; Sadilek, A.; Abumoussa, A.; Borrello, D.; Kushalnagar, R. S.; and Bigham, J. P. 2012. Real-time captioning by groups of non-experts. In Miller, R.; Benko, H.; and Latulipe, C., eds., *The 25th Annual ACM Symposium on User Interface Software and Technology, UIST '12, Cambridge, MA, USA, October 7-10, 2012*, 23–34. ACM.

Le, J.; Edmonds, A.; Hester, V.; and Biewald, L. 2010. Ensuring quality in crowdsourced search relevance evaluation: The effects of training question distribution. In *In SIGIR 2010 workshop*, 21–26.

Liu, Q.; Peng, J.; and Ihler, A. T. 2012. Variational inference for crowdsourcing. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States.*, 701–709.

Naim, I.; Gildea, D.; Lasecki, W. S.; and Bigham, J. P. 2013. Text alignment for real-time crowd captioning. In Vanderwende, L.; III, H. D.; and Kirchhoff, K., eds., *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, 201–210. The Association for Computational Linguistics.

Nie, W. 2000. Waiting: integrating social and psychological perspectives in operations management. *Omega* 28(6):611 – 629.

Nushi, B.; Singla, A.; Gruenheid, A.; Zamanian, E.; Krause, A.; and Kossmann, D. 2015. Crowd access path optimization: Diversity matters. In *Proceedings of the Third AAAI Conference on Human Computation and Crowdsourcing, HCOMP 2015, November 8-11, 2015, San Diego, California.*, 130–139.

Roy, S. B.; Lykourentzou, I.; Thirumuruganathan, S.; Amer-Yahia, S.; and Das, G. 2015. Task assignment optimization in knowledge-intensive crowdsourcing. *VLDB J.* 24(4):467–491.

Sarma, A. D.; Parameswaran, A. G.; and Widom, J. 2016. Towards globally optimal crowdsourcing quality management: The uniform worker setting. In *Proceedings of the 2016 International Conference on Management of Data, SIGMOD Conference 2016, San Francisco, CA, USA, June 26 - July 01, 2016*, 47–62.

Sheng, V. S.; Provost, F. J.; and Ipeirotis, P. G. 2008. Get another label? improving data quality and data mining using multiple, noisy labelers. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, USA, August 24-27, 2008*, 614–622.

Simmons, R. J. 2010. Profile luis von ahn: Recaptcha, games with a purpose. *ACM Crossroads* 17(2):49.

Trestian, R.; Moldovan, A.; Muntean, C. H.; Ormond, O.; and Muntean, G. 2012. Quality utility modelling for multimedia applications for android mobile devices. In *IEEE international Symposium on Broadband Multimedia Systems and Broadcasting, BMSB 2012, Seoul, Korea (South), June 27-29, 2012*, 1–6.

Voice of America. 1942. VOA News.

von Ahn, L., and Dabbish, L. 2005. ESP: labeling images with a computer game. In *Knowledge Collection from Volunteer Contributors, Papers from the 2005 AAAI Spring Symposium, Technical Report SS-05-03, Stanford, California, USA, March 21-23, 2005*, 91–98. AAAI.

von Ahn, L. 2013. Duolingo: learn a language for free while helping to translate the web. In Kim, J.; Nichols, J.; and Szekely, P. A., eds., *18th International Conference on Intelligent User Interfaces, IUI '13, Santa Monica, CA, USA, March 19-22, 2013*, 1–2. ACM.

Wang, Y.-Y.; Acero, A.; and Chelba, C. 2003. Is word error rate a good indicator for spoken language understanding accuracy. In *IEEE Workshop on Automatic Speech Recognition and Understanding*.

Wang, J.; Ipeirotis, P. G.; and Provost, F. J. 2017. Cost-effective quality assurance in crowd labeling. *Information Systems Research* 28(1):137–158.

Zhang, J.; Shiraishi, Y.; Kumai, K.; and Morishima, A. 2016. Real-time captioning of sign language by groups of deaf and hard-of-hearing people. In *Proceedings of the 18th International Conference on Information Integration and Web-based Applications and Services, iiWAS 2016, Singapore, November 28-30, 2016*, 54–63.