Process-Based Quality Control Techniques in Crowdsourcing

by

Ria Mae Harina Borromeo

A dissertation submitted in partial fulfillment for the
degree of Doctor of Philosophy in Engineering

to the
School of Science for Open and Environmental Systems
Graduate School of Science and Technology
Keio University

August 2017

Abstract

Crowdsourcing has become a popular approach to complete tasks that are tedious using manual methods or difficult for automatic ones. As crowdsourcing taps the capacities of humans, its possibilities are endless. However, the unpredictability of human behavior and the actuality of human error make it difficult to consistently achieve high-quality outcomes, posing quality control as a major challenge in crowdsourcing. Although many strategies have been proposed to optimize data quality, their effectiveness is dependent on each particular crowdsourcing application. To solve this, more techniques and experiments from which humans and algorithms can learn from are needed to be able to build a recommendation system that proposes quality management techniques depending on the attributes of the crowdsourcing application.

In this dissertation, I approach quality management in crowdsourcing based on the sub-processes involved, specifically: task design, task deployment, and task assignment. I first experimented on factors affecting task design. In particular, I tested the effect of task complexity on a data extraction task and crowd type on a sentiment analysis task. Experiments show that there is no significant difference in the quality achieved from simple and more complex versions of a data extraction task and that the performance of paid unpaid workers are comparable in a sentiment analysis task. For task deployment, task deployment strategies were proposed along three dimensions: work structure, workforce organization, and work style. To semi-automatically implement these strategies in a crowdsourcing platform, a deployment tool was designed and developed. The effectiveness of the strategies when applied to text creation tasks were then studied and recommendations were drafted for both crowdsourcing researchers and practitioners. Finally, for task assignment, a fuzzy clustering-based method for building a personalized summary of tasks, also known as composite tasks, for crowd workers was validated. As observed from the experiments, personalization improves the workers' overall experience and that diversifying tasks can improve the workers' output quality.

Author: Ria Mae Harina Borromeo
Supervisor: Motomichi Toyama

Acknowledgements

One day when I was around eight years old, my sisters and I received a balloon as a souvenir from a birthday party. While playing at home, I wondered, *what if we light our balloon with fire? Will it pop or simply melt?* I shared the idea with my sisters, and my eldest sister agreed to light the balloon using a match. After that, we found out the answers to my questions. As the balloon contained Hydrogen, a flammable gas, lighting it with fire will make the balloon explode and burn things around it such as children's faces. Thanks to that experience, my ideas got more logical, and I learned to plan my experiments better. Nevertheless, I still had crazy ideas growing up, and I am extremely grateful to my mom (Teresita), dad (Emerlito), and two sisters (Katrina and Elaiza), for humoring me and supporting my ideas, especially my "craziest idea" of pursuing a Ph.D.

This pursuit would not have been possible without my supervisor, Prof. Motomichi Toyama who has guided and supported me ever since I was Masters student. Thank you always, Sensei. I would also like to thank Sihem Amer-Yahia for involving me in her research projects, mentoring me in the process. I am also grateful to the members of my thesis committee: Prof. Issei Fujishiro, Prof. Kyoko Ohara, Prof. Yoshihisa Shinozawa, and Prof. Shingo Takada for reviewing my research and helping me improve it.

Next, I would like to thank Thomas Laurent, Maha Alsayasneh, Ayushya Anand, Shady Elbassuoni, Anas Hossami, Vincent Leroy, Julien Pilourdault, Manish Singh, and Anna Stavrianou, with whom I worked with on various projects. Thank you also to paid crowd workers and volunteers for participating in our experiments.

I would also like to thank Manuel Vergel and Mark Richard Velasco for giving me programming advice, my labmates for cheering me on, and all my other friends, especially Maria Kennison, Maris Catinding, Maylee Fontaine, Aiko Matsumoto, Sharie Echegoyen, Katrina Soliman, and Makoto Okuyama for believing with me.

Thank you as well to Keio University and Japan's Ministry of Education, Culture, Sports, Science, and Technology (MEXT) for granting me a full scholarship, enabling me to research full-time.

Finally, I would like to thank the Lord Jesus Christ. Everything I have, I have received from Him. To Him be the glory.

Contents

List of Figures

List of Tables

For You.

Chapter 1

Introduction

Computer technologies, which were once just used to aid individuals and organizations in their operations, have become a necessity. Due to the ability of computers to perform high speed and exact calculations, many human processes have been automated. Nevertheless, humans still perform better than computers in areas such as ideation, judgment, and perception. To maximize the strengths of both humans and computers, *human computation*, a computer science technique that is involved in the design or analysis of information processing systems in which humans participate as computational elements [10], has been widely studied.

Crowdsourcing is a form of human computation defined as the practice of obtaining information or services by soliciting input from a large number of people via the Internet [5]. It has become a popular solution to complete tasks that are tedious using manual methods or difficult for automatic ones.

The term *Crowdsourcing* was first published in 2006 in Jeff Howe's Wired Magazine article entitled *The Rise of Crowdsourcing* [62]. He further defined it as the act of taking a job traditionally performed by an employee and outsourcing it to an undefined, generally large group of people in the form of an open call [4]. Since then, varying definitions of crowdsourcing have emerged. In 2012, Estellés-Arolas et. al attempted to come up with an integrated definition of crowdsourcing. They analyzed 40 original definitions from research papers in the databases of ACM, IEEE, ScienceDirect, SAGE, and Emerald, and came up with a definition that covers any crowdsourcing initiative. Their definition is as follows: *Crowdsourcing is a type of participative online activity in which an individual, an institution, a* *non-profit organization, or company proposes to a group of individuals of varying knowledge, heterogeneity, and number, via a flexible open call, the voluntary undertaking of a task. The undertaking of the task, of variable complexity and modularity, and in which the crowd should participate bringing their work, money, knowledge and/or experience, always entails mutual benefit. The user will receive the satisfaction of a given type of need, be it economic, social recognition, self-esteem, or the development of individual skills, while the crowdsourcer will obtain and utilize to their advantage what the user has brought to the venture, whose form will depend on the type of activity undertaken* [47]. From their definition, we can identify the following elements of crowdsourcing.

- **Requester**: an individual, an institution, a non-profit organization, or a company that has a problem to be solved or tasks to be completed
- **Task**: the work to be done which is of variable complexity and modularity
- **Crowd**: a group of individuals of varying knowledge, heterogeneity, and number who provides solutions or completes tasks; also known as workers
- **Platform**: an application that provides crowdsourcing functionalities related to crowd and task management
- **Incentive**: satisfaction of a given type of need, be it economic, social recognition, self-esteem, or the development of individual skills
- **Solution**: what the user has brought to the venture

Several sub-processes are involved in the crowdsourcing process: task design, task deployment, task assignment, and answer aggregation. First, the requester designs a task then deploys it in a crowdsourcing platform. In the crowdsourcing platform, workers choose tasks to work on or are automatically assigned tasks to complete. Workers then complete the tasks. Once answers have been received, the platform or the requester aggregates the answers.

A popular crowdsourcing example is the search for Jim Gray. Gray was a computer scientist who received a Turing Award for his seminal contributions to database and transaction processing research and his technical leadership in system implementation [11]. Early in 2007, he failed to return from his sailing trip around the Farallon Islands. One of the efforts done to find him and his boat was capturing satellite images of the ocean then asking volunteers from the general public to identify which images should be further examined. His colleagues posted tasks in the Amazon Mechanical Turk [1] (AMT) that asked volunteers to compare a few 300 by 300 pixels image sub-tiles to a template then provide a score for each sub-tile for evidence of features similar to the ones provided in a given template [60]. In this example, the requester is Gray's colleagues; the task is to find images that should be further examined; the crowd consists of AMT workers; the platform is AMT, and the solution is the aggregate of results received for each sub-tile. There is no explicit incentive or extrinsic reward, but the workers were likely driven by an intrinsic motivation to help.

1.1 Problem

As crowdsourcing taps on the capacities of humans, its possibilities are endless. For example, crowdsourcing applications are used in traffic management to collect data regarding traffic conditions in specific geographic locations [7], in medicine to help diagnose difficult medical cases [3], in education to assist in the grading and reviewing of homework [2], and in behavioral research to conduct user studies [99]. Due to its success, crowdsourced solutions are continuously being developed. However, since crowdsourcing relies on humans, the unpredictability of human behavior and the actuality of human error make it difficult to consistently achieve high-quality outcomes. Furthermore, according to Kittur et al., workers tend to *satisfice* or minimize the amount of effort on their part and in the extreme cheat or game the system [72].

In the previous example, if one worker accidentally missed sighting Jim Gray's boat in the images and another one did not diligently examine the photos, without quality control mechanisms, the results from the crowd could be incorrect.

Hence the question, *how do we control quality in crowdsourcing?* While quality control can be a subjective, models and metrics to quantitatively and objectively assess quality along different dimensions of a software system, have been proposed [18]. Examples of these dimensions include reliability, accuracy, relevancy, completeness, and consistency [14]. In crowdsourcing, quality can be defined as the *extent to which an outcome conforms to the requirements of the requester* [18].

[1] https://www.mturk.com/

Efforts to answer the question of quality control in crowdsourcing come in different forms. However, since the question is very general, I break it down according to the processes involved in crowdsourcing and ask more specifically, *how do we control quality in task design, task deployment, task assignment, task completion, and answer aggregation?*

Examples of existing answers include the following. In *task design*, one can adhere to best practices that include task decomposition, providing simple and clear instructions, testing tasks, and providing fair wages [96, 119]. In *task deployment*, one can choose between parallel and iterative workflows [89]. In *task assignment*, one can use tasks recommender systems to help workers find tasks they like [129]. In *task completion*, one can employ micro-breaks, such as a game, to allow workers to relax during long sequences of tasks and potentially improve their performance [114]. Lastly, in *answer aggregation*, one can get multiple answers and select which one the majority chooses [65].

Nevertheless, while specific approaches have proven to be effective for particular applications, they cannot be generalized for all task types. For instance, using an *iterative workflow* wherein a worker builds on the output of another worker was found to be effective for tasks such as writing image descriptions, brainstorming company names, and transcribing blurry text [89]. However, for a taxonomy creation task, using an iterative workflow does not yield positive results because the taxonomy grows with every iteration thus making tasks more time-consuming and overwhelming [42].

It is indeed challenging to propose generic strategies for optimizing output quality due to crowdsourcing's versatility and broad application domains. To address this, more techniques and experiments from which humans and algorithms can learn from are needed to be able to build a recommendation system that proposes quality management techniques depending on the attributes of the crowdsourcing application [18].

1.2 Contributions

To contribute to quality control crowdsourcing research, quality control techniques in crowdsourcing were first classified according to the sub-processes they are involved in: task design, task deployment, task assignment, task completion, and answer aggregation. After that, techniques were proposed, and experiments were conducted for the first three sub-processes. The following enumerates this research's contribution.

1. *Experiments on Design Factors in Crowdsourcing.* In particular, experiments were conducted to test the effect of the task complexity on a data extraction task and crowd type on a sentiment analysis task. Experiments show that there is no significant difference in the quality achieved from simple and more complex versions of a data extraction task, and that paid and unpaid workers perform similarly in a sentiment analysis task.
2. *Deployment Strategies for Crowdsourcing Text Creation.* Task deployment strategies were proposed along three dimensions: work structure, workforce organization, and work style. A tool was developed to enable the strategies' semi-automatic deployment in a crowdsourcing platform. The strategies' effectiveness when applied to text creation tasks was investigated, and recommendations for crowdsourcing researchers and practitioners were presented.
3. *Personalized and Diverse Task Composition in Crowdsourcing.* For task assignment, a fuzzy clustering-based method for building a personalized summary of tasks (composite tasks) for crowd workers was validated. Experiments show that personalization improves the workers' overall experience and that diversifying tasks can improve the workers' output quality.

1.3 Dissertation Overview

The dissertation is organized as follows. Chapter 2 reviews existing work in quality control in crowdsourcing. Chapter 3 explains the experiments I performed to test the effect of crowd type on sentiment analysis and task complexity on data extraction. Chapter 4 presents the proposed deployment strategies applied to text creation tasks. Chapter 5 presents the proposed technique for task assignment and details the experiments that were performed to validate it. Chapter 6 summarizes our contributions and concludes.

1.4 The Use of “We” Instead of “I”

The reader might notice that in Chapters 4 and Chapters 5, the pronoun, “we” is used instead of “I.” This is because my contributions will not be possible apart from the research team where I belonged to.

Chapter 2

Related Work

This chapter discusses in detail the sub-processes in crowdsourcing and the current quality control techniques that are being practiced in each sub-process.

2.1 Task Design

Task design involves planning the work that the requester wants to outsource to the crowd. It is similar to the process of preparing an exam for students to answer or software for people to use. The requester needs to think about various factors such as the task definition, instructions, user interface, complexity or granularity of the task, incentives and compensation policy, and type of crowd to employ. We explain some of these factors in the following.

Task Complexity. Task complexity is related to the cognitive dimensions of a task that can be apparent in task design and can, in turn, affect outcome quality [111]. Yang et al. emphasized that the characteristics of the task itself are crucial in the optimization of the crowdsourcing process [128]. They derived a unique regression model which automatically predicts the complexity of the task, eliminating the need of “piloting" it first. The correct estimation of the complexity of the task, which is subjective and relies both on the task and task doers, can help determine a fitting approach and a suitable reward strategy for the task, thereby increasing the output quality [128].

Since superior performance and outputs have been observed when tasks are short and explicit (micro-tasks), than when the tasks are complex, Kittur et al. implemented *CrowdForge* to decompose compound tasks into simpler subtasks using the MapReduce programming paradigm combined with human intelligence [73]. Kulkarni et al. further improved the system with *Turkomatic*, which utilizes a recursive work flow where tasks are divided into elementary sub-tasks and solved, in multiple cycles [78].

Kittur et al. also pointed out that a poorly designed task is also a key reason for a poor output in crowdsourcing tasks [72] because while the task may seem fairly clear to the requester, it might not be the same for others. In addition, a simple task with an uncomplicated interface and clearly stated reward strategies are likely to improve the quality of output [50].

Incentives. Payment rates and schemes have also been identified to affect quality. Mason et al. deployed two types of tasks: an image ordering task using different wage rates, and a puzzle where workers were asked to find words hidden in an array, under either a quota or piece-rate payment scheme, and different wage rates. From both experiments, they reported that higher payments increased the quantity of work but not the quality; and that payment scheme can have a significant effect on quality [100]. Mao et al. adapted a volunteer-based citizen science project from Planet Hunters that asks the crowd to annotate or identify light curves, into AMT and used different payment schemes to pay the workers: pay per task, pay for time, pay for annotation. They observed that for the same amount of money, different payment schemes result in significant differences in the quality of results [95]. Likewise, Finnerty et al. implemented several rewards based on various motivations for a handwriting recognition task. The rewards the used were *none* (no extra reward), *please* (the crowd was requested to do the task quicker or more accurately), *fixed* (given a fixed amount), and *dynamic* (based on their performance). They found that the dynamic reward led to better results when applied to both time and accuracy simultaneously.

Crowd Type. There are many ways to classify the crowd, one of which is according to the type of incentive they receive. In such case, we can distinguish paid workers from unpaid workers or volunteers. Many studies have tried to explain the effect of incentives on the quality of the output. Workers in an unpaid crowdsourcing task are driven by different motives than the workers in paid tasks. Unpaid crowd workers or volunteers are people who are enthusiastic about the work and thus may have knowledge about the domain or are casual workers who are participating for leisure [95]. On the other hand, workers in a paid system have a financial incentive for completing the task as their prime motivation.

Volunteer-based crowdsourcing such as Sarah Parcak's *Global Xplorer*, for example, asks users to tag significant locations in a pre-processed satellite imagery of various archaeological sites [8]. The project has been able to identify over thousands of important sites which demonstrate the success of a volunteer based crowdsourcing task.

When workers are solely motivated by the financial incentives of the task, they are likely to avoid the cognitive effort required to complete the task and produce a bare minimum [95], unless there are evaluation and quality assessment strategies in place [75]. It has also been observed that increasing financial incentives, may improve the quantity [100] and the latency [86] of output but not the quality [36, 59, 75].

2.2 Task Deployment

Task deployment involves implementing your design. It can be further divided into two parts: deployment planning and actual deployment. Sometimes deployment planning may be integrated into task design, but in this review, they are separated. In task deployment, a requester considers factors such as the crowdsourcing platform to use, the work structure, the workflow type, and the work style. Some of these factors are discussed in the following.

Crowdsourcing Platforms. A crowdsourcing platform is a system that connects requesters and workers. It is commonly a web application that provides functionalities for task and crowd management. For requesters, the choice of a crowdsourcing platform may depend on the nature of the projects they want to crowdsource and the incentive they are willing to provide to the workers. Some crowdsourcing platforms are specialized in particular projects e.g. InnoCentive[1] for

[1] https://www.innocentive.com/

research and development and ClickWorker [2] for managing e-Commerce data while other platforms are general-purpose such as AMT, microWorkers[3], and CrowdFlower[4].

General-purpose platforms usually specialize in handling simple tasks or microtasks. Examples of microtasks include labeling, transcription, text tagging, and sentiment analysis. However, there are more complex tasks that are context-heavy, interdependent, require more cognitive effort, and may take many hours to complete [57]. These tasks are called macrotasks. Examples of macrotasks include programming, document editing, and sentence translation. Macrotasks may be deployed on general purpose platforms in their original form or decomposed into microtasks. Larger tasks with more complex requirements such as software engineering and journalism are typically published in generic online outsourcing marketplaces or global online work platforms such as Upwork [5] and Freelancer[6].

Work Structure. Work structure refers to how workers are organized when they collaborate. Therefore it is important to consider the varied dispositions of the crowdsourcing workers in these different collaboration schemes. Andre et al. observed that when there are multiple workers working together, various psychological factors come into play [24]. They primarily categorize these factors as motivational such as *social loafing*, *evaluation apprehension* and *sucker effects*, or coordinational such as *production blocking* and *thought derailment.*

They also highlighted the following from previous studies: the quality of the output is a function of both task structure and the task type and that implementing simultaneous task structure has been found better when the task is uncertain and interdependent, requiring workers to interact, while sequential task structure is suitable when the task is of creative nature where workers build upon the output of other workers [24].

Workflow Type. Little et al. [89] have explored the idea of performing tasks in an iterative or parallel manner. They reported that performing tasks iteratively, where one person builds upon another's output, results in a better average output

[2] https://www.clickworker.com/
[3] https://ttv.microworkers.com
[4] https://www.crowdflower.com/
[5] https://www.upwork.com/
[6] https://www.freelancer.com/

quality in a variety of tasks. While it is better to execute the task in parallel when diversity is required in the output, the marginal utility of each worker decreases when the crowd size increases. With larger groups, other complications arise such as efficiently aggregating the final output of all the crowd workers. On the other hand, iterative task workflow also suffers from a serious drawback where an error committed by even one worker might be magnified in the later iterations by other workers.

Ambati et al. also concluded that their *pipeline work flow*, where workers enhance the output of previous workers in three phases, yields a superior result than the traditional crowdsourcing work flow in translation tasks [20].

Work Style. Researchers have also tried to combine algorithmic intelligence in crowdsourcing to further optimize the process. Work style distinguishes a hybrid approach, wherein both algorithms and humans are combined to complete a task, from a crowd-only approach, wherein the task is completed solely by the crowd. Fan et al. noted that though tasks such as knowledge discovery, annotation, and schema matching can be performed using purely machine-based approaches, involving the crowd, which is inherently better at analyzing semantic correspondences between different entities than machines, can perform the same task effortlessly and with better accuracy [49]. They implemented a hybrid machine and crowdsourcing approach to solve the web table schema matching problem.

A related application called *CrowdER* implements the task of entity resolution using hybrid crowdsourcing [123]. Entity resolution is similar to schema matching in that it requires finding semantic correlations among objects, and hence is a suitable task to crowdsource. *CrowdER* attempts to reduce the number of microtasks, which would otherwise be very large, by employing a two-tiered approach involving machines. It implements algorithms to filter out the easily discernible objects and only crowdsource the complex ones, thereby reducing both latency and cost of the crowdsourcing process [123].

2.3 Task Assignment

Task assignment is a process in crowdsourcing wherein workers are matched to tasks and vice versa. In the agent coalition domain, task assignment involves organizing agents to complete a task that cannot be completed by a single agent. Studies focus on the formation of a group of agents that would be able to complete the tasks most efficiently [17, 83, 116]. In crowdsourcing, there are two main types of task assignment methods: push and pull. In push methods, the crowdsourcing platform assigns tasks to workers while in pull methods, workers find tasks via a user interface and self-appoint themselves to tasks [84].

Using pull methods, a worker needs to search appropriate tasks from a list of tasks, which can be difficult especially when the list is huge. This can reduce overall throughput, accuracy, and engagement in a crowdsourcing system [87]. High search costs threaten to reduce the motivation to participate and cause workers to settle for less suitable tasks which can decrease the quality of the results [53]. Efforts to address this include task recommendation frameworks that consider workers' preferences have been proposed. Ambati et al. modeled workers preferences based on a worker's profile, task metadata, and feedback from both workers and requesters and used a bag-of-word scheme to calculate similarities between tasks [21]. Yuen et al. based their model on a worker's work performance history and task searching history [129]. Lin et al. considered implicit signals about task preferences such as types of tasks that have been available and have been displayed, and the number of tasks workers select and complete [87]. Using these frameworks, available tasks are shown to workers.

Pull methods, which are popular on volunteer-based crowdsourcing platforms such as Zooniverse[7] and Crowd4u[8], directly assign appropriate tasks to workers when they arrive [38]. These methods typically consider different factors such as workers' reputation, task evaluation metrics, and human factors.

Given a fixed set of tasks and a budget that specifies the number of times each task should be completed, the *Dual Task Assigner (DTA)* algorithm learns workers' skills through exploration and builds on the idea of online primal-dual formulation [61]. The *iCrowd* framework estimates a worker's skill level by inferring the accuracy of a worker's answer based on her performance on similar tasks that have already been completed [48]. In cite [68], a worker's reliability is learned by comparing one worker's answers to others.

[7] https://www.zooniverse.org/

[8] http://crowd4u.org/

Instead of learning the workers' skills, the *QASCA* framework incorporates task evaluation metrics such as accuracy and f-scores into the assignment strategies [131]. More specifically, the framework examines combinations of tasks and estimates the quality improvement if a combination is assigned to an incoming worker. The combination with the potential maximum quality improvement is then selected.

Human factors have also been considered in task assignment. In collaborative crowdsourcing, where workers work together in groups to accomplish a task, Rahman et al. studied how to form a group who could most effectively work together to complete an assigned task [110]. They found that considering the comfort level of workers who work together (affinity) improves quality. Roy et al. modeled task assignment as an optimization problem and propose adaptive algorithms that consider human factors such as worker expertise, availability, and wage requirement [113]. For a collaborative news document editing task, their framework achieves high-quality and efficient task assignments within a specific budget. Pilourdault et al. proposed an adaptive algorithm that considers the worker's motivation [107]. They modeled motivation as a balance between the difference in skills required by the tasks and the reward amount. They found that assigning relevant tasks to workers has a positive impact on task throughput while assigning tasks that balance task diversity and payment obtain the highest output quality.

Given these research efforts, Kittur et al. pose a research question: should tasks be pulled by workers or pulled by platforms [72]?

2.4 Task Completion

Task completion is the process wherein workers complete the tasks that have been assigned to them. Previously discussed crowdsourcing processes directly affect this process. For example, increasing financial reward in task design has been found to raise the amount of work done by the crowd, which leads to faster completion times [100]. Employing a parallel work structure [89] in task deployment and using a system that dynamically assigns tasks to different workers to meet real-time demands as in [37] can also lead to faster completion times.

Essentially, requesters want their tasks to be completed fast by good workers while workers complete tasks depending on their motivations. Existing work on this mainly falls on motivating workers to participate in crowdsourcing to minimize the latency of task completion.

According to Deci and Ryan, motivations can be either intrinsic or extrinsic. Intrinsic motivation exists when an individual is activated because he finds fulfillment in an activity while extrinsic motivation exists when an individual performs an activity to achieve something else [43]. Kaufmann et al. further categorized these motivations into enjoyment-based motivation, community-based motivation, immediate payoffs, delayed payoffs, and social motivation. They found that although extrinsic motivation categories such as immediate payoffs (payment), delayed payoffs (human capital advancement), and social motivation, affect how much time a worker spends working, intrinsic motivation aspects are more important [69].

Various techniques have been proposed to increase workers' motivation. For instance, gamification or integrating game mechanics into a design attempts to increase the crowd's intrinsic motivation to perform an activity or change behavior has been shown to have a positive impact in crowdsourcing particularly in participation [102]. Priming or the temporary implicit activation of behavioral tendencies as a result of exposure to an environmental stimuli has been found to be able to lead to significant performance gains [101]. Furthermore, Chandler et al. found that framing tasks in a meaningful manner improves crowd participation. The use of micro-breaks that allow workers to relax during long sequences of tasks also appeared to increase overall worker engagement and work commitment [114]. Providing requester feedback during task completion has also been shown to improve the quality of crowd work [45].

Aside from those techniques, Bigham et al. proposed *quikTurkit*, an abstraction layer on top of *Turkit* [90] API that re-posts tasks to keep them visible. It also recruits workers before they are needed and keeps them busy by asking them to solve previously asked questions [28]. Bernstein et al. then developed techniques to recruit synchronous crowds. In one of their models, workers are given a small reward for staying on call then alerted when work becomes available [27]. Another tool called *ReLauncher* does not provide additional motivation for workers to speed up task completion. Instead, during runtime, it identifies tasks that other workers have left unfinished and relaunches them for other workers to complete [77]. The *CLAMShell* system, which focuses on data labeling, speed up crowds by to minimizing latency in the different stages of labeling.

2.5 Answer Aggregation

After workers complete tasks, answers often need to be aggregated into a specific form. This process can be challenging due to the difference in the workers' characteristics. There may be diligent and skilled workers, but there may also be dishonest workers who give random answers and unskilled workers who give poor quality answers. Workers may be biased by their preferences or misunderstanding of the tasks [124]. Furthermore, subjective and creative answers are more difficult to aggregate. An effective approach to answer aggregation is the employment of expert reviewers who manually check the contributions from the crowd. While this is reliable, its scalability is low, and the cost is high.

Automatic methods have been proposed for answer aggregation. Hung et al. surveyed aggregation techniques in crowdsourcing [65]. Techniques can be categorized into iterative and non-iterative techniques. Iterative techniques leverage mutual reinforcing relations between workers and answers while non-iterative techniques compute the aggregated answer as a linear combination of votes [103].

Majority consensus is one of the most commonly used non-iterative techniques. The final answer is decided based on the majority vote over the total number of answers required for a question. A variation of majority voting is used in the CrowdFlower platform implements a weighted majority voting wherein they assign weights to the answers of the workers based on a worker's confidence score [9].

Output agreement is another common aggregation technique wherein an answer is deemed correct if a specific number of workers independently and simultaneously provide the same answer to a question [18]. This was done in an image labeling game where the goal is to correctly label images. Players independently label the same image and a label is accepted if a player's label matches the label provided by another workers [121].

Having a *gold standard dataset* also helps in answer aggregation. A *gold standard dataset* is a set of questions, each with a known correct answer. When answers are known, crowd answers can straightforwardly be compared to them and answers that achieve a specified level of similarity or accuracy can be accepted as a correct answer. However, not all tasks can have gold answers and most of the time gold answers are difficult to create. In [104], the authors proposed a mechanism, currently being used in the CrowdFlower platform, to programmatically generate gold standard datasets. Although the method relies on manual spot checks to detect worker errors, it has been shown to improve the quality and scalability of crowd-sourced data collection. Nevertheless, it cannot be applied to highly subjective tasks.

In conjunction with other aggregation methods, techniques for detecting and filtering spam responses are also commonly employed. These techniques can be simple such as checking the amount of time a worker took to complete [46, 71] and removing those that were done too fast. Answers can also be directly labeled as spam and filtered out if they do not match the gold standard set. A more sophisticated approach is determining whether a worker is a spammer or not based on his work history then filtering out answers from a worker labeled as a spammer. To detect spammers, Expectation Maximization-based algorithms [66, 122] and Machine Learning techniques [58, 80] have been explored. While these approaches have been found to be effective, they have only been tested on specific tasks.

Chapter 3

Experiments on Design Factors in Crowdsourcing

3.1 Background

Designing crowdsourcing tasks requires careful planning since a large number of factors, such as task complexity, crowd type, and incentives, impact the quality of task outcomes. While specific parameter settings for these factors have proven to be effective for specific applications, they cannot be generalized for all task types. Let's take incentive design as an example. Mason et al. reported that in an image ordering task using different wage rates, and a puzzle where workers were asked to find words hidden in an array, higher payments increased the quantity of work but not the quality [100]. However, Aker et al. found that for tasks that involve math and general questions, providing higher payment positively impacts the quality of results [16]. It is important to understand how design factors impact different task types to be able to control them effectively. Thus, I attempted to validate previously studied design factors on new task types. This chapter presents an investigation on the effects of controlling crowd type and task complexity on sentiment analysis and data extraction tasks respectively.

For each experiment, I will discuss how the task design, task deployment, task completion, and answer aggregation process were conducted.

3.2 Crowd Type on Sentiment Analysis

There are many ways to classify the crowd, one of which is according to the type of incentive they receive. In such case, paid workers can be distinguished from unpaid workers or volunteers. Paid crowd members receive a monetary incentive in exchange for completing tasks. They are typically recruited in paid crowdsourcing platforms such as AMT, CrowdFlower, and microWorkers. By contrast, unpaid crowd members receive no monetary incentive. They are possibly users who were required to perform tasks or asked to volunteer [44]. They are typically enthusiastic about the work and may have knowledge about the domain or are casual workers who are participating for leisure [95]. The unpaid crowd can contribute through unpaid crowdsourcing platforms such as Crowd4U, Zooniverse, and Crowdcrafting[1]. While the paid crowd can be easier to recruit, tapping the unpaid crowd can be an economical alternative for requesters who are concerned about the budget [35].

Related studies have different findings regarding the performance of paid and unpaid crowds. In Mao et al.'s work where they adapted an annotation task that was originally performed by volunteers in the Planet Hunters citizen science project, to an experiment in AMT with paid workers [95]. They investigated how three types of payment schemes (pay per task, pay for time, and pay per annotation) influenced the behavior of paid workers compared to volunteers. Their findings show that given appropriate incentives, paid crowd workers might work at a faster rate and achieve similar accuracy compared to volunteers who are working on the same task [95]. Goncalves et al. compared the performance of unpaid situated crowdsourcing for counting Malaria-infected blood cells with the performance of the same task deployed in AMT. They observed that in unpaid crowdsourcing through public displays, the accuracy of results was lower, but the rate of uptake of tasks was higher compared to AMT [55].

Since related studies are still inconclusive, I conduct experiments to investigate further the quality of results produced by paid and unpaid workers. In this particular experiment, I assess the impact of crowd type on the more subjective task of sentiment analysis.

[1] https://crowdcrafting.org/

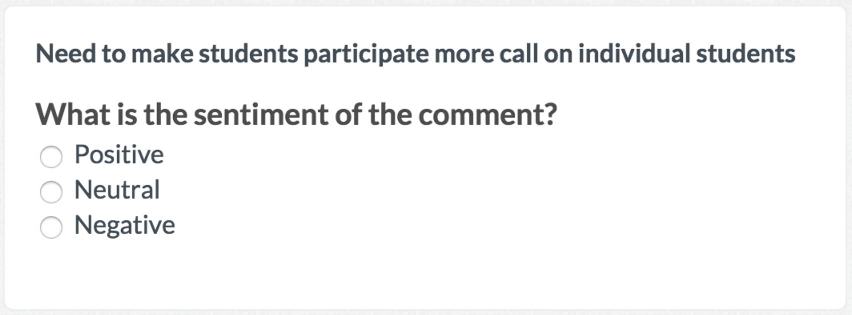
Sentiment analysis entails interpreting the feeling conveyed by a message. The most basic formulation of this problem is to determine if a message is positive or negative. Although automatic sentiment analysis has made much progress in recent years, humans still are more effective than algorithms at interpreting ambiguous messages such as sarcasm for example. This ambiguity makes sentiment analysis tasks perfectly suited to crowdsourcing.

This experiment consists of sentiment analysis of student evaluation comments. The comments were collected from the Student Evaluation of Teaching (SET) comments for Professor Jonathan Cox from 2006 to 2012 [12]. Professor Jonathan Cox is a Mathematics professor at the State University of New York at Fredonia. From the unstructured document that contained the comments, 418 comments with an average number of 46.52 words each and a standard deviation of 35.48 were extracted. The sentiment analysis consisted in determining if a student evaluation was negative, neutral or positive. Details of the experiments are explained in the following subsections.

3.2.1 Task Design

The task displays a student evaluation comment and asks the worker, “ *What is the sentiment of this comment?*” Workers can choose from three answers: “*positive*”, “*neutral*” and “*negative*”. I set an incentive of $0.01 for each comment to be analyzed and requested three answers per comment.

3.2.2 Task Deployment



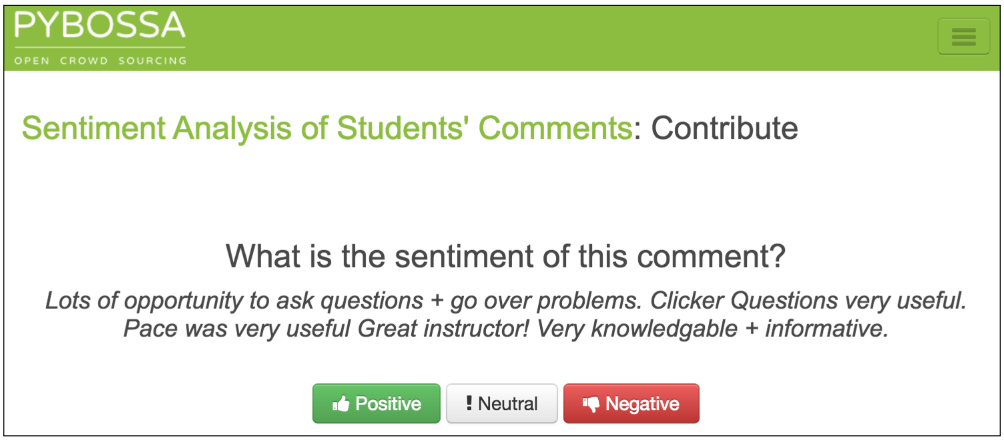
Need to make students participate more call on individual students

What is the sentiment of the comment?

○ Positive
○ Neutral
○ Negative

Figure 3.1: Sentiment analysis task in CrowdFlower

To reach the paid crowd, I deployed the task on CrowdFlower, a paid crowdsourcing platform that specializes in distributing small, discrete tasks to many



Sentiment Analysis of Students' Comments: Contribute

What is the sentiment of this comment?

Lots of opportunity to ask questions + go over problems. Clicker Questions very useful. Pace was very useful Great instructor! Very knowledgable + informative.

FIGURE 3.2: Sentiment analysis task in PyBossa

online workers in an assembly line fashion [120]. An example of the task as deployed on CrowdFlower can be seen in Figure 3.1.

To reach the unpaid crowd, I deployed the task in a private instance of PyBossa, an open-source volunteer crowdsourcing platform [13]. I then advertised the task in my social network. An example of the task as deployed in PyBossa is shown on Figure 3.2.

3.2.3 Task Completion

The paid version was completed in 2.90 hours by 86 workers. The crowd consisted of 86 identified workers from CrowdFlower's pool of workers that were gathered through different channels. Each worker performed an average of 14.58 tasks with a standard deviation of 7.64. They came from at least 28 countries, and the countries with the most number of workers are India, Bosnia and Herzegovina, and Romania.

In the unpaid version, 46 volunteers completed all the tasks in 44.8 hours. Each volunteer performed an average of 28.50 tasks with a standard deviation of 60.65. They came from 5 countries, and the country with the most number of workers is the Philippines.

3.2.4 Answer Aggregation

The output from the paid crowd was obtained in an aggregated form directly from CrowdFlower. For every task, CrowdFlower chooses the response which has the highest confidence score. The confidence score is based on a worker's trust score, a value that ranges from 0 to 1, where 0 is the lowest and 1 is the highest. The confidence score is calculated by adding the trust scores of workers then dividing it by the sum of trust scores of workers for a specific task [9]. The summary generated by CrowdFlower showed that the paid crowd detected 141 positive, 231 negative, and 46 neutral comments.

The output from the volunteers was retrieved in raw format. Post processing had to be performed to derive the sentiment of each comment. I adopted CrowdFlower's formula in deriving the final judgment. However, since the volunteers' trust scores were not available, I assumed that they were all trustworthy and assigned them a trust score of 1. The resulting formula is equivalent to the rule of the majority. When a task received three different responses, it was classified as neutral. There were 138 positive, 204 negative, and 76 neutral comments derived from the unpaid crowd's responses.

3.2.5 Results

In Table 3.1, the comparison of the two methods in terms of crowd cost, completion time, agreement and accuracy is summarized. *Crowd cost* is the amount paid to the platform and the crowd workers. *Completion time* refers to the time from when the project was launched to the time when all required responses were received. *Accuracy* is the percentage of similarity of the results compared to a gold standard. I used the manual evaluation of the same comments in [34] as the gold standard to measure the methods' accuracy. Lastly, *Agreement* is the degree to which the results agree with the gold standard. The strength of agreement is calculated by deriving Cohen's kappa coefficient, a statistic which measures inter-observer agreement of qualitative data. In this study, each method is assumed to be an independent observer.

The calculation of the Kappa statistic is based on the difference between the observed agreement or how much agreement is actually present, compared to how much agreement would be expected to be present by chance alone. The coefficient lies on a scale of -1 to 1 where 1 denotes a perfect agreement, 0 means chance agreement and negative values indicate agreement less than chance such as disagreement between observers [117]. A coefficient of 0.01 to 0.20 indicates slight agreement, 0.21 - 0.40 indicates fair agreement, 0.41 - 0.60 indicates moderate agreement, 0.61 - 0.80 indicates substantial agreement, and 0.81 - 0.99 indicates an almost perfect agreement [81].

TABLE 3.1: Sentiment analysis results from paid and unpaid Crowds

	Paid	Unpaid
Crowd Cost (USD)	15.35	0.00
Completion Time (hours)	2.90	44.80
Accuracy	76.08	76.32
Agreement	0.577	0.597

As outlined in Table 3.1, results from the paid crowd were obtained significantly faster than the unpaid method. In terms of quality (accuracy compared to the gold standard), the two methods achieved very similar results. Regarding the agreement to the gold standard, the agreement coefficient of the unpaid version is slightly higher than that of the paid method. However, the coefficients' interpretations are the same: both methods moderately agree with the gold standard.

3.3 Task Complexity on Data Extraction

The concept of task complexity is related to the cognitive dimensions of a task, which can be used in task design and in turn affect performance [40]. Previous work on categorization and annotation tasks found that simpler tasks led to better results [50, 95]. I thus verify if limiting task complexity is as effective for ensuring outcome quality when crowdsourcing data extraction tasks.

Data extraction tasks involve retrieving specific data from unstructured raw data. These kinds of tasks are typically difficult to automate since the input data may require perception or may be in a unparsable format such as an image. The viability of crowdsourcing as an approach to data extraction has been explored in various studies. In [92], Lofi et al. showed the relevance of crowdsourcing to data extraction tasks and explored different tasks where crowdsourcing provides a solution whereas automatic extraction or full manual extraction fail or are prohibitively expensive. The authors also showed that a hybrid approach combining crowdsourcing and automated data extraction outperformed other approaches and encouraged further investigation of these techniques. In [88], Ling et al. also harnessed the power of the crowd to extract parallel translation data from the social network Twitter, showing better results than with previous approaches. Seeing how effective a tool crowdsourcing is for data extraction, I chose to use this field as an experiment subject.

In this experiment, I asked workers to extract specific information from a digital archive of research paper presentations. An entry in the digital archive contains PowerPoint and PDF files written in either English or Japanese. There are currently 341 entries in the digital archive, but in this project, entries from 2008 to 2009 were chosen as the sample set. The sample set consists of 23 Japanese entries and 46 English entries.

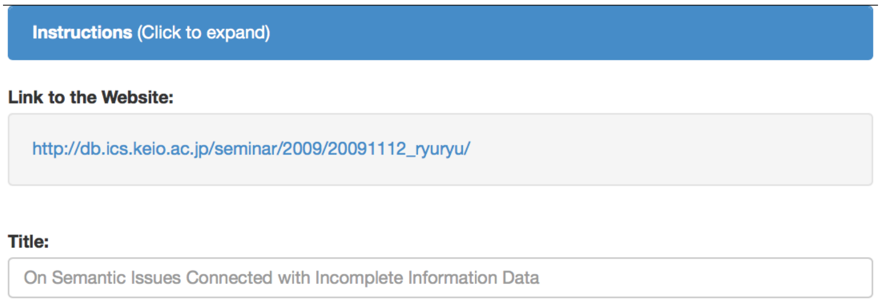
3.3.1 Task Design

The data extraction task provides a link to a digital archive of research paper presentations and an input form. It instructs the worker to open the link and extract the specified information. As the digital archive included documents in Japanese, carefully-designed instructions were provided to allow workers to complete the task without Japanese proficiency.

To explore the influence of task complexity on the quality of the extracted data, two versions of the task were designed. The first one, the simple version, is shown on Figure 3.3. In this version of the task, the workers were asked to extract a single entity, the paper title, from the archive entry. The second version or the complex version is shown on Figure 3.4. In this version, the workers were asked to extract six entities from an archive entry. This task design required the user to examine more parts of the entry and to extract different types of data, such as strings and dates, making the task more complex.

3.3.2 Task Deployment

The tasks were deployed in AMT and workers were asked to extract information from 69 entries in the digital archive. For each entry, both versions of the data extraction tasks, requesting three answers per entry were published. For the simple



Instructions (Click to expand)

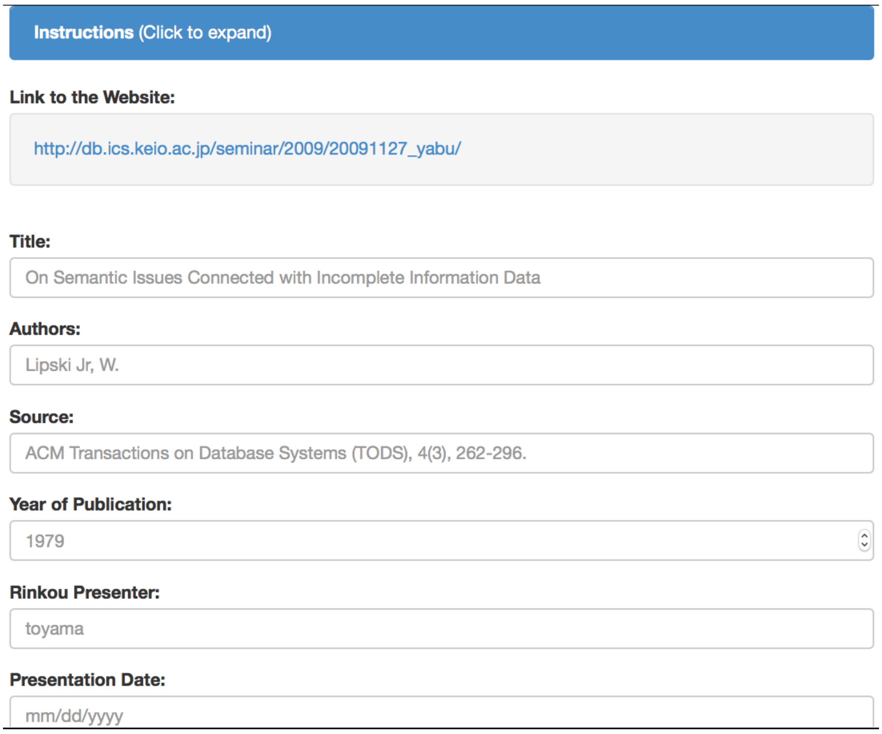
Link to the Website:

http://db.ics.keio.ac.jp/seminar/2009/20091112_ryuryu/

Title:

On Semantic Issues Connected with Incomplete Information Data

Figure 3.3: Simple data extraction task



Instructions (Click to expand)

Link to the Website:

http://db.ics.keio.ac.jp/seminar/2009/20091127_yabu/

Title:

On Semantic Issues Connected with Incomplete Information Data

Authors:

Lipski Jr, W.

Source:

ACM Transactions on Database Systems (TODS), 4(3), 262-296.

Year of Publication:

1979

Rinkou Presenter:

toyama

Presentation Date:

mm/dd/yyyy

Figure 3.4: Complex data extraction task

version $0.02 per task was given as a reward while for $0.12 was provided for the complex version.

3.3.3 Task Completion

The simple version was completed by 20 workers in one hour and 38 minutes while the complex version was completed by 30 workers and took 3 hours and 35 minutes to complete.

3.3.4 Answer Aggregation

To be able to evaluate the quality of work produced by the crowd, a gold standard set was constructed by manually extracting the titles of the papers, querying Google Scholar for their corresponding bibliographic information, then storing the results in a file. Then a script that extracts the presenter and presentation date from entry's URL, and the publication source and year from the results of the Google Scholar queries was created.

The raw results consisted of 3 answers per entry. To select the best entry, the forced agreement method, wherein at least two matching answers are considered the correct answer, was used. When no answers matched, the first one was selected.

3.3.5 Results

I compared the final answers to a gold standard. For each field, I computed the similarity of the worker submission to the gold standard using the PHP function *similar_text*, which outputs the similarity percentage of two strings as described in Programming Classics: Implementing the World's Best Algorithms by Oliver [106]. I then averaged the similarities of each entry's submission's fields to obtain an answer's accuracy, which was used as the quality metric.

Table 3.2: Average accuracy per language and per task version

Language	**Number of items**	**Simple**	**Complex**
Japanese	23	66.6%	60.57%
English	46	98.14%	96.67%
Total	69	88.54%	85.69%

The average output accuracy for each task version is listed in Table 3.2. The accuracy of the Japanese and the English entries of the archive are also specified in case the language had an influence. Furthermore, the number of items in each language is also reported to show the influence of each category to the overall result.

A two-tailed paired t-test on the accuracy of results from the simple and complex tasks was conducted. The calculated t-value was -0.66235 and the p-value was 0.508871. Based on these values, it can be inferred that the two data sets are not significantly different at $p < 0.1$.

The results show that although the average accuracy of results (English and Japanese combined) from the simple task is higher than that obtained by the complex task, this difference is not statistically significant.

3.4 Discussion

In this section, I will discuss some threats to the validity of our experiments. According to Lee and Borgo, there is no repeatability in crowdsourcing studies [25]. It is widely known that recruiting a different crowd is likely to lead to different results. Indeed, in crowdsourcing, the crowd poses the most significant threat to validity. Since we are dealing with humans, many factors can threaten the validity of the study. However, I will focus on factors that affect who becomes members of the crowd and consequently affect the results.

- **Time of the day when the task was deployed** - Since about 80% of AMT workers are from the United States and about 20% are from India [67], we requesters have access to more workers if we deploy task at times when people in the US are working. Although I was not able to measure the quality of workers at different times of the day, I noticed that task completion time was faster when experiments were deployed in AMT in the morning in Japan compared to when deployed in the afternoon.
- **Social network** - When recruiting volunteers from one's social network such as in Section 3.2, the characteristics of one's social network affects who gets recruited. In this case, I had access to more than a thousand people from which 75% are from the Philippines. Since Filipinos are highly active in social media [41], I was able to recruit enough workers to complete the tasks. This could be different if the cultural composition of my social network were different.
- **Required qualifications** - In AMT, this filters the workers who are allowed to do tasks I deployed. I required workers to have at least 90% task acceptance rate to work on the tasks. In CrowdFlower, I only allowed workers with *Level 1* qualification as it means that they have completed over a hundred test questions across a variety of task types, and have a very high

overall accuracy. Setting these parameters differently could lead to different results.

- **Incentives** - Since some workers are driven by the monetary reward to work, the amount of monetary incentive affects who gets recruited.
- **Crowdsourcing platform status** - In AMT, as tasks are displayed in lists, workers might only browse the first few pages thus when there are many tasks posted in AMT at a given time, some tasks which are in the next pages might recruit fewer workers.

Furthermore, in the sentiment analysis task, it was observed that results from both paid and unpaid crowds are similar and both moderately agree to the gold standard. While the effect of crowd type on quality is not evident in this experiment, it is possible that slightly changing a parameter setting could achieve different results. Let's take for example the content of the comments to be analyzed. Since the comments are made by students, and easily relatable, most workers can understand their underlying sentiment. However, if the topic is for example about jeepneys as public transportation, workers who have never ridden jeepneys may not be able to perceive the real sentiment of the comment.

A similar data extraction experiment was performed in [31] where the tasks were deployed in the CrowdFlower platform with lower rewards ($0.005 for the simple and $0.03 for the complex). The findings were also different as simple tasks achieved significantly more accurate results [31]. Intuitively, it was surmised that simple tasks would always yield higher quality results. However, in this experiment, it was observed that comparable quality could be achieved by complex tasks when other parameters such as reward and platform of choice are altered.

3.5 Chapter Summary

In this chapter, I presented experiments on crowd type and task complexity, two design factors that impact quality. In these experiments, no significant difference in the quality achieved from simple and more complex versions of a data extraction task was observed. Furthermore, it was noted that the performance of paid and unpaid workers are comparable in a sentiment analysis task.

Chapter 4

Deployment Strategies for Crowdsourcing Text Creation

4.1 Background

Crowdsourcing has been applied to all kinds of tasks ranging from the simplest such as image categorization to more sophisticated ones such as the creation of elaborate text. Although several automatic solutions have been designed for text creation, this task remains difficult for machines as it involves a level of abstraction and creativity that only humans currently possess. Text creation is also challenging for humans because it requires comprehension and edition, two time-consuming operations. That is particularly true for translation, summarization, and narrative writing where inputs of varying length and complexity need to be understood to proceed with a task. I, as part of a research team, therefore explored deployment strategies for crowdsourcing text creation tasks to improve the effectiveness of the crowdsourcing process. Our team considered effectiveness through the quality of the output text and the cost of deploying the task, and the latency in obtaining the output. We formalized a deployment strategy in crowdsourcing along three dimensions: work structure, workforce organization, and work style. Work structure can either be simultaneous or sequential, workforce organization independent or collaborative, and work style either by humans only or by using a combination of machine and human intelligence. We implemented these strategies for translation, summarization, and narrative writing tasks by designing a semi-automatic tool that uses the Amazon Mechanical Turk API and experimented with them in different input settings such as text length, number of sources, and topic popularity. In this chapter, I will explain our proposed strategies and experiments, and report our findings regarding the effectiveness of each strategy. Lastly, I will present recommendations to guide requesters in selecting the best strategy when deploying text creation tasks.

4.2 Deployment Strategies

We define a deployment strategy as a plan on how to carry out a task. It is a combination of three dimensions: *work structure*, *workforce organization*, and *work style*. Work structure refers to how a task is deployed among workers, which can either be *simultaneous* or *sequential*. Workforce organization refers to how workers are organized to complete a task; it can either be in an *independent* or *collaborative* fashion. Work style distinguishes a *hybrid* approach, where a task is completed by both algorithms and humans, from a *crowd-only* approach, where a task is solely carried out by humans. The combination of those dimensions results in 6 strategies. In the following sub sections, the three dimensions and the resulting strategies will be discussed.

4.2.1 Deployment Dimensions

4.2.1.1 Work Structure

Work structure refers to the way in which tasks are distributed, which can either be *sequential* or *simultaneous*. Using a sequential work structure, a task outcome is passed to the next worker to be improved, whereas, in a simultaneous structure, several outcomes are produced in parallel from the same initial input.

Surveying existing work, we found that while general-purpose crowdsourcing platforms such as AMT mainly support a simultaneous (i.e. parallel) completion of independent tasks, Little et al. introduced a sequential work structure that involves an iterative workflow paradigm wherein a worker builds on or evaluates the work of another worker [90]. They implemented TurKit, a toolkit that deploys iterative tasks on AMT. TurKit employs a fixed policy that performs improvement tasks until it consumes a given budget. Studies that compare sequential and simultaneous work structures suggest that the recommended work structure depends on the type of task. For tasks such as writing image descriptions, brainstorming company names, and transcribing blurry text, Using a sequential work structure has been found to improve average response quality [89]. Similarly, for a limerick writing task, Andre et al.'s findings reveal the sequential work structure to be more effective as the number of workers collaborating on a task increases [24]. However, for a taxonomy creation task, using sequential work does not yield positive results because the taxonomy grows with every iteration thus making tasks more time-consuming and overwhelming [42]. The sequential work structure also does not fare better for an outline creation task where a tournament workflow, which allows multiple merges of independent parts of an outline in parallel, produces faster, higher quality and more diverse outlines [94].

4.2.1.2 Workforce Organization

Workforce organization refers to how workers are organized to complete tasks, which can either be in an *independent* or a *collaborative* fashion. In a collaborative organization, a worker collaborates with others at a specific time (simultaneously) to complete a task and produce one result. In an independent one, a worker completes a task alone and outputs one result.

This dimension focuses on determining the appropriate set of workers for a specific task. Simple tasks such as labeling an image and judging the sentiment of text, are commonly done by workers independently. However, previous studies show that more complex tasks such as translation [20], workflow design [78], user interface control [82], and article writing [73], are more effectively done by workers collaborating together. Appropriately assigning workers to collaborate on a task, however, is a challenge.

We also found another way to organize a workforce in existing works, and it is based on the known quality of workers. *CrowdFlower* assigns levels to workers based on their work history. Requesters can specify the required level for workers to be able to complete their tasks. *RABJ* maintains a tiered worker hierarchy enabling workers to be assigned to tasks based on their performance [74]. *MobileWorks* hires managers, a particular class of workers who are in charge of recruiting new workers, evaluating potential problems with requester-defined tasks, and resolving task discrepancies [79]. *Argonaut* uses a predictive model of worker quality to select qualified workers as reviewers of others' work [57].

4.2.1.3 Work Style

Work style distinguishes a *hybrid* approach, where the task is completed by both algorithms and humans, from a *crowd-only* approach, which is solely carried out by humans.

Combining algorithms and humans in crowdsourcing has also been previously explored. Crowdsourcing database systems such as Qurk [97], Deco [105], and CrowdDB [51] combine relational database management systems and crowdsourcing. They follow the basic workflow of query processing, which involves parsing the query, generating one or more query plans, then selecting the best query plan using both humans and machines [86]. The ability of the crowd to provide results to queries that traditional database systems cannot answer, such as those that involve subjective comparisons and unknown or incomplete data, complements the known strengths of database systems.

Another common approach combines automatic methods and crowdsourcing to reinforce each other. For instance, in a structured data extraction task, the Argonaut system uses automated extractors and machine learned classifiers to identify the components of a document then asks reviewers from the crowd to correct the output of the automated extractors [57]. In a sentiment analysis of reviews, various machine learning algorithms were used to classify reviews [126]. If the classifications produced by the algorithms disagree for a particular review, the review is assigned to humans then the results from the algorithms and crowdsourcing are aggregated to derive the outcome. For a web table matching task, *CrowdWeb* introduces a concept-based approach that maps each column of a web table to the best concept in a well-developed knowledge base and asks the crowd to discern the concepts that are difficult to discern automatically [49].

4.2.2 Resulting Strategies

Combining the dimensions discussed in the previous section, we define six deployment strategies described in Table 4.1 and illustrated in Figure 4.1. We do not

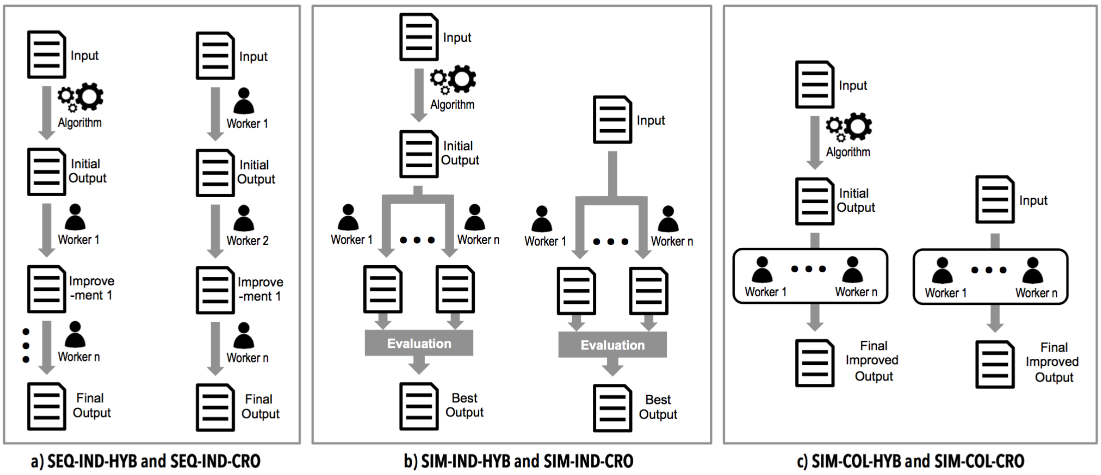


FIGURE 4.1: Deployment strategies

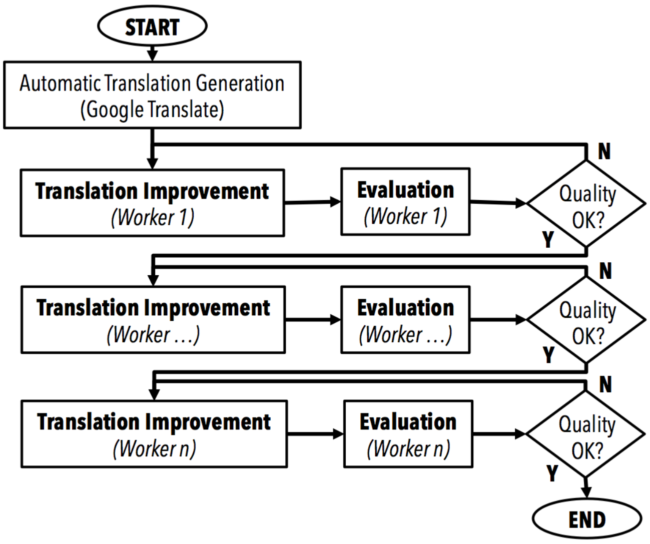

START
Automatic Translation Generation (Google Translate)
Translation Improvement (Worker 1)
Evaluation (Worker 1)
Quality OK?
N
Y
Translation Improvement (Worker ...)
Evaluation (Worker ...)
Quality OK?
N
Y
Translation Improvement (Worker n)
Evaluation (Worker n)
Quality OK?
N
Y
END

FIGURE 4.2: Translation: SEQ-IND-HYB process flow

consider the combination of sequential work structure and collaborative workforce organization as we define collaboration to be simultaneous.

To further illustrate our deployment strategies, let us examine a translation task example. In addition to the original text, user interface, and task instructions, a requester must consider the following: the number of workers to recruit for the task and the required result quality, which are both affected by time and budget constraints. Suppose we have an article in French to be translated into English as our input and we want to get results with the highest possible outcome quality that three workers can achieve within no particular time frame and without budget restrictions. If we use SEQ-IND-HYB, the process flow will be as in Figure 4.2. We first generate an initial translation from French to English using, for example,

Table 4.1: Deployment strategies

Strategy	**Description**
Sequential-Independent-Hybrid (SEQ-IND-HYB)	An initial output is generated automatically then it is sent to one worker at a time for improvement. The final result is a single output.
Sequential-Independent-CrowdOnly (SEQ-IND-CRO)	An initial output is completed by a worker then it is sent to one worker at a time to improve it. The final result is a single output.
Simultaneous-Independent-Hybrid (SIM-IND-HYB)	An initial output is generated automatically then sent to several independent worker for improvement. The best output is chosen after an evaluation.
Simultaneous-Independent-CrowdOnly (SIM-IND-CRO)	Several outputs are created simultaneously by independent workers. The best output is chosen after an evaluation.
Simultatneous-Collaborative-Hybrid (SIM-COL-HYB)	An initial output is generated automatically then sent to one group of workers who collaborate to improve it.
Simultaneous-Collaborative-CrowdOnly (SIM-COL-CRO)	One output is created by one group of workers together.

Google Translate[1], then ask three workers to improve the translation one after the other.

Since we want the highest possible quality, we evaluate every response received. The evaluation may be done by experts, by algorithms [93], or by the crowd [39].

Different parameters such as the number of improvements or reward amount, and evaluation methods (automatic, expert, or crowdsourced), potentially affect the results and could be further explored later on. However, such considerations are out of this study's scope as we focus on the effect of the deployment strategies only.

4.3 Deployment Tool

Deploying crowdsourcing tasks manually is a challenging and time-consuming process, especially when more complex workflows are involved. Although several tools

[1] https://translate.google.com/

and frameworks have been proposed to address this issue, such as Turkit [90], Automan [26], and several others explored in [76], they are designed for different applications. Thus, we developed a tool, *CDeployer*, to semi-automatically deploy our proposed strategies. We describe the use of *CDeployer* and make it available on github[2] so that it can help requesters deploy their tasks more easily or serve as a basis for other initiatives. Please see A for details on how to use this tool.

Figure 4.3 presents a deployment workflow using our tool. To explain further, suppose we want to deploy a summarization task using SIM-IND-CRO. First, through the Command Line Interface (CLI), the requester issues a `make CreationTask` command with the text to be summarized and a configuration file that contains information such as AMT credentials, and the task template as input. The deployment tool then processes the inputs and publishes a Human Intelligence Task (HIT) in AMT through API calls. Workers choose HITs to work on and complete them. When a HIT is completed, the requester is notified by email. The requester can review the HIT using the deployment tool, which automatically rejects an answer if it matches an automatically generated text. If no HITs have been automatically rejected, the requester can manually approve all tasks, and then he can issue a `make EvaluationTask` command that asks other workers to rate the summaries. Once that task completes, the tool can be used to aggregate the ratings and output the best summary.

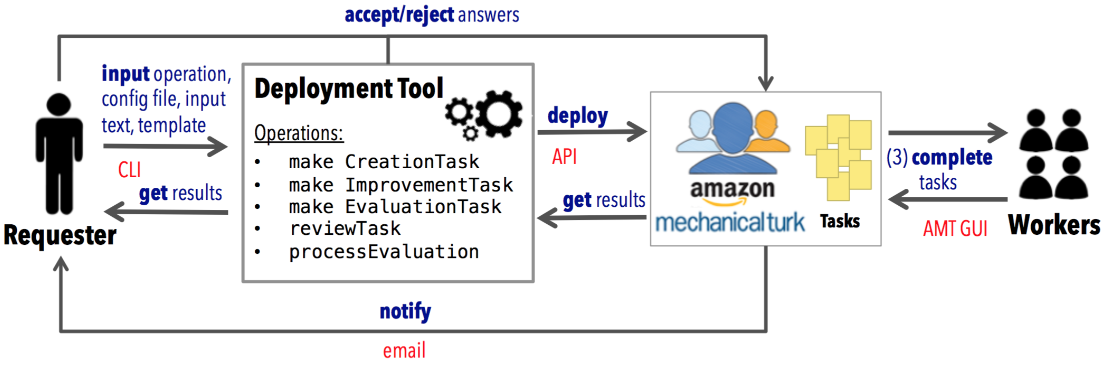


Figure 4.3: Workflow using *CDeployer*

4.4 Text Creation Tasks

We are interested in three text creation tasks: translation, summarization, and narrative writing. It has been shown that for translation, letting workers edit

[2] https://github.com/riamaehb/CDeployer

text and correct each others' mistakes in a sequential manner produces higher quality translations compared to when workers generate independent translations simultaneously [24]. In the case of summarization, it has been shown that automatic methods are not very good at summarizing and merging sentences to generate high-quality summaries [115], and while there are existing tools that can effectively generate narratives, the resulting texts are inferior compared to human generated ones [108]. We further discuss related studies on text creation tasks in the following subsections.

4.4.1 Summarization and Narrative Writing

Text summarization and narrative writing require to process information, either textual or in another form, to create a text, either shorter or more human-friendly than the source. Summarization has been extensively studied and automatic tools such as MEAD [109] are now available. However, these automatic methods do not generate high-quality summaries, as shown in [115]. Crowdsourcing and crowd-hybrid methods [64] have been used to produce higher quality summaries. In [91] the authors use crowdsourcing to identify patterns in the way humans generate summaries. In [70], the authors developed a prototype called Storia that crowdsources writing stories summarizing posts on social media. They observed that users were more engaged with stories showing a narrative structure and were also more likely to recommend them to others. In [108], the authors design an automatic tool that generates narratives from medical data about newborns. They find that while their system produced effective summaries of the clinical data, the resulting texts were poor compared to human-generated ones.

4.4.2 Translation

Similarly to text summarization, various machine translation (MT) algorithms, and tools exist. MT algorithms may be rule-based, statistical, example-based, or hybrid [23]. Recently, crowdsourcing has been used in language translation where workers are commonly tasked to collect training data for statistical machine translation methods or to translate text. Zaidan et al. solicit redundant translations and select the best one based on the output's score. The scores are based on objective features of both the translator and the translation such as country of residence, the number of years speaking English, edit rate from other translations, and optional calibration against professional translators [130]. Yan et al. propose a two-step process that includes translation and post-editing by non-professionals. They also collect redundant translations and use a graph-based algorithm that considers the collaborative relationship between translators and editors to select the best translation [127]. Instead of editing translations from the crowd, Aikawa et al. investigate the impact of post-editing machine translations produced by Microsoft Research's Collaborative Translation Framework (CTF). The editors, who were students, were able to translate a university's website to 9 languages within two months [15].

We apply the deployment strategies to three text creation tasks: translation, summarization, and narrative writing. For translation, we study the impact of text length by examining two types of translation tasks: long text (LTT) and short text translation (STT). For summarization, we study the influence of the number of sources to be summarized and consider two different summarization tasks: single source summarization (SSS), where a single text input is summarized, and multi-source summarization (MSS), where several independent input texts are summarized into one. Lastly, in narrative writing, we look at the impact of the subject's popularity and compare results on a popular topic (PNW) and a less popular one (UNW).

4.5 Experiments

In this section, we describe the experiments we performed to evaluate our proposed strategies and report our results and findings. We deployed the three types of text creation tasks described in Section 4.4, using the strategies presented in Section 4.2.2 through *CDeployer*, the tool we developed, described in Section 4.3. For hybrid strategies, we used Google Translate[3] to obtain machine translations, MEAD[4] to automatically produce automatic summaries, and a template text to generate narratives. Due to the unpredictable nature of the crowdsourcing market, we were unable to run conclusive collaborative experiments using these tasks. We thus exploit results obtained in a previous round of experiments closely related to this study published in [30].

[3] *http://translate.google.com*

[4] *http://www.summarization.com/mead/*

We observed how our strategies affected the cost and result quality, and latency of crowdsourced text creation tasks. We calculated the *cost* by taking the sum of all the payments to workers for all the HITs posted to carry out a strategy.

To evaluate the *quality*, each output was rated for fluency and adequacy, similar to machine translation evaluation methods [125]. Fluency and adequacy were rated according to a five-point Likert scale (1 - very poor, 2 - poor 3 - barely acceptable, 4 - good, 5 - very good) as was practiced in [56]. The rating was performed by experts who were unaware of the underlying workflows. Specifically, texts in summarization and narrative writing tasks were evaluated by a graduate student in computer science with a Test of English for International Communication (TOEIC) score of 990 while translations were evaluated by a student in an English graduate program in Computer Science, whose native language is French.

We attempted to measure *latency* by adding the amount of time it took for a worker or group of workers to complete each task in a given strategy. However, we only present anecdotal findings on latency as we observed that it primarily depends on the availability of willing and qualified workers. Additionally, Huang et al. reported that latency is also affected by the time of day at which the task is posted [63].

4.5.1 Setup

4.5.1.1 Tasks

Translation. To observe the impact of text length, we performed LTT and translated one long press article entitled, *La Joconde sourit car elle est heureuse, tout simplement*, and STT using a shorter article entitled, *Grande Barrière de corail : scientifiques et militants en appellent à l'Unesco*, from French to English. The former consists of 385 words while the latter consists of 87 words.

Summarization. We summarized movie reviews from the IMDB data set[5] for the movie, *The Matrix*. We examined two different summarization tasks: SSS where we asked for a summary of a 399-word review and MSS where we requested

[5] https://www.kaggle.com/orgesleka/imdbmovies

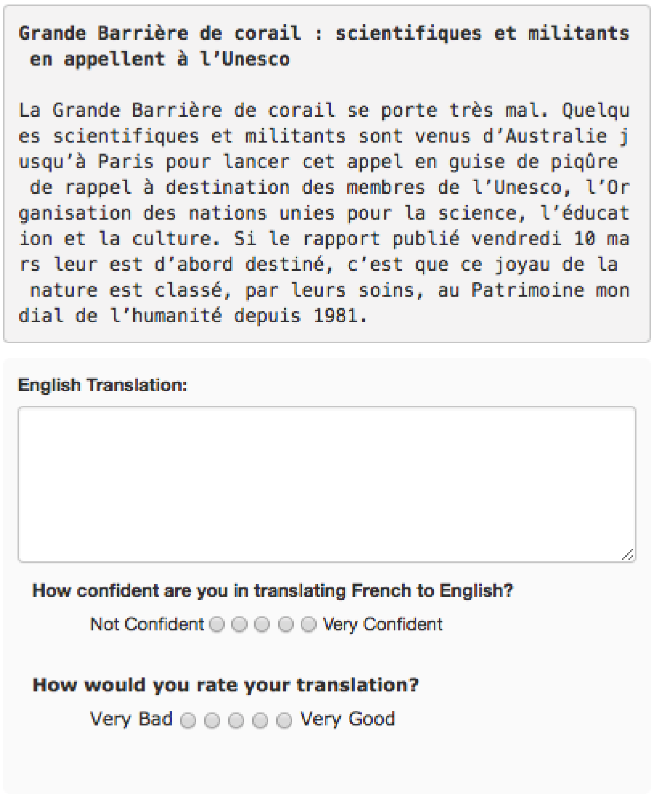
three movie reviews with an average of 165 words per review, to be summarized into one.

Narrative Writing. In this task, the goal was to write a narrative in English based on tabulated statistics of soccer matches from `whoscored.com`. We examined two different narrative writing tasks: PNW with a match between Real Madrid and Las Palmas, and UNW with a match between Alavez and Sevilla.

4.5.1.2 Strategies

Table 4.2: HIT type rewards in USD

	Translation		Summarization		Narrative Writing	
	Long	**Short**	**Single-source**	**Multi-source**	**Popular**	**Not Popular**
Creation	2.00	0.75	2.00	2.00	1.00	1.00
Improvement	0.66	0.25	0.66	0.25	0.33	0.33
Evaluation	0.75	0.30	0.75	0.75	0.45	0.45



Grande Barrière de corail : scientifiques et militants en appellent à l’Unesco

La Grande Barrière de corail se porte très mal. Quelques scientifiques et militants sont venus d’Australie jusqu’à Paris pour lancer cet appel en guise de piqûre de rappel à destination des membres de l’Unesco, l’Organisation des nations unies pour la science, l’éducation et la culture. Si le rapport publié vendredi 10 mars leur est d’abord destiné, c’est que ce joyau de la nature est classé, par leurs soins, au Patrimoine mondial de l’humanité depuis 1981.

English Translation:

How confident are you in translating French to English?

Not Confident ○ ○ ○ ○ ○ Very Confident

How would you rate your translation?

Very Bad ○ ○ ○ ○ ○ Very Good

Figure 4.4: Creation task for short text translation

Table 4.3: HIT type composition per strategy

HIT Type	**SEQ-IND-CRO**	**SEQ-IND-HYB**	**SIM-IND-CRO**	**SIM-IND-HYB**	**SIM-COL-CRO**	**SIM-COL-HYB**
Creation	1		m		1	
Improvement	n	n		m		1
Evaluation			p	p		
Invitation					1	1
Total	**n + 1**	**n**	**m + p**	**m + p**	**2**	**2**

We deployed the strategies described in Section 4.2.2 by using four types of HITs: creation, improvement, evaluation, and invitation HITs. A *creation HIT* asks a worker to create text from scratch and rate the quality of his work. An improvement HIT presents a worker with an initial text that he is asked to rate and improve. An *evaluation HIT* shows a worker three text outputs, which he is asked to rate in terms of fluency and adequacy. An *invitation HIT* invites workers to collaborate on a HIT and work together simultaneously.

As indicated in Table 4.3, the SEQ-IND-CRO strategy is made of one creation HIT and n improvement HITs. SEQ-IND-HYB, on the other hand, is only made of n improvement HITs. SIM-IND-CRO is made of m creation HITs and p evaluation HITs. In the evaluation HIT, each worker is asked to rate the fluency and adequacy of m texts. The ratings are then aggregated to get the best outcome. Instead of a creation HIT, SIM-IND-HYB is composed of m improvement HITs and p evaluation HITs. The collaborative strategies both have an invitation HIT coupled with a creation HIT in SIM-COL-CRO, and an improvement HIT in SIM-COL-HYB. Figure 4.4 shows a sample of a creation HIT for short text translation and Table 4.2 lists the rewards we set for each HIT type.

In our experiments, we set n to 2 and set m and p to 3. These values were chosen as we consider them to be the minimum values that allow observable effects of the strategies on the output. However, we do not study the effects of each of these variables on the output and leave this analysis for future work.

4.5.2 Results

In this section, we report our findings based on the experiments we performed. For every task type, the quality of text produced by the strategies is shown in

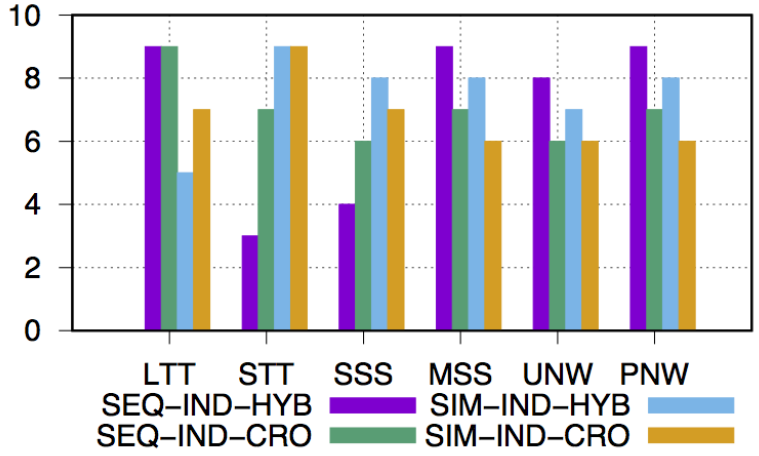


FIGURE 4.5: Resulting quality of strategies per task type

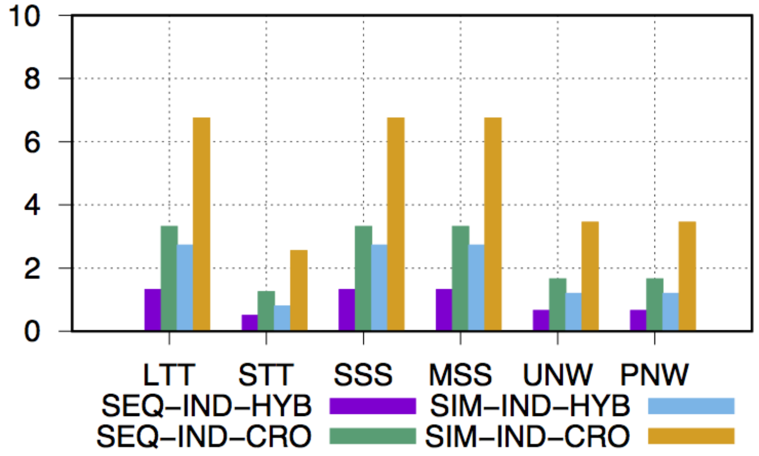


FIGURE 4.6: Cost of strategies per task type

Figure 4.5. The quality is a sum of a 5-point adequacy rating and 5-point fluency rating. The cost of these strategies, which is a sum of the rewards of the HITs in a strategy, is shown in Figure 4.6. Although we do not observe an obvious quality trend, there are constant trends in cost: sequential strategies are cheaper than simultaneous ones, and crowd-only strategies are more expensive than hybrid ones. As previously mentioned, we do not have concrete findings on latency as we found it to be dependent on many factors such as time of day and human error. However, theoretically, simultaneous work structures complete faster than sequential ones as they can be executed in parallel, while sequential structures must be executed one after the other. We will discuss more specific findings for every task type in the following subsections.

4.5.2.1 Translation

Work Structure (SEQ vs. SIM). For long texts, our experiments show the superiority of a *sequential* work structure over a *simultaneous* one. *Sequential* strategies allow workers to iteratively improve translations, which overall leads to the best results for both work styles. In addition, as workers do not start from scratch, their workload is smaller which reduces overall cost. In terms of latency, *sequential* plans take longer to complete compared to *simultaneous* plans since the start time of a task depends on the completion of the previous task. For short texts, however, having several translations as in *simultaneous* plans, enables evaluators to choose high-quality text.

Work Style (HYB vs CRO). Using a *hybrid* approach significantly reduces the cost and latency of translating a long text. Workers are given an initial translation to improve, which requires less work than starting from scratch. For both long and short texts, we do not find that machine translation negatively affects the quality of results, as workers can correct errors of the initial translation and achieve good quality. Using a *crowd-only* approach, we can get both good and bad translations as we cannot predict workers' honesty and diligence when working alone. Hence, we do not observe any benefit from fully human-crafted translations. Also, many workers tend to use automatic online translation tools, even when explicitly asked not to.

Workforce Organization (IND vs COL). Based on the experiments in [30], we found that when using a *hybrid* approach, workforce organization does not affect result quality, but controls the trade-off between cost and latency. A *collaborative* organization increases latency, as workers spend time discussing and synchronizing their work. However, as each of them performs a fraction of the work, the total cost decreases. Conversely, an *independent* organization requires less time but costs more. In the case of a *crowd-only* approach, workforce organization has a significant impact on the quality of results. Workers are used to performing micro-tasks that require a short attention span and limit the risk of working for a long time and not being granted a reward by the requester. Translating long texts from scratch takes time, so an *independent* organization leads to poor results, as workers tend to rush their work at the expense of quality. Furthermore, working as a group for translation tasks seems to have a positive impact on the behavior of workers which also contributes to raising translation quality. Another advantage of a *collaborative* organization is a much lower cost.

4.5.2.2 Summarization

Work Structure (SEQ vs SIM). In Figure 4.5, we can see that the *sequential* work structure is superior to the *simultaneous* one for multi-source summarization, while it is the opposite for single source summarization. We then examined the improvements done in *sequential* work structure and found that the revisions were mostly syntactical and did not alter the content of the text. The task was then more of an editorial task, rather than a real improvement task. This made the quality of the outcome highly dependent on the initial summary. The *simultaneous* work structure may be more expensive but since improvements in the *sequential* work structure were minor, the *simultaneous* work structure was able to provide more options wherein the best outcome could be selected from.

Work Style (HYB vs. CRO). For algorithms, MSS is more complex than SSS as there are more factors to consider such as redundancy of information, temporal dimensions in the sources, and compression ratio [54]. Indeed, *crowd-only* approaches for MSS performed better than their *hybrid* counterparts. In SSS, the *crowd-only* and *hybrid* approach produced an outcome of comparable quality.

Workforce Organization (IND vs. COL). In [30], we found that when summarizing movie reviews, workers *collaborate* efficiently, which leads to better results than an *independent* organization on all evaluation criteria. However, getting workers to work together at specific times is a challenge particularly in crowdsourcing marketplaces where workers operate on different timezones.

4.5.2.3 Narrative Writing

Work Structure (SEQ vs. SIM). Although summarization and narrative writing tasks are similar, we found that for topics such as soccer, the improvements done to initial texts were not just on syntax but also on content. This makes the *sequential* work structure effective. Strategies using the *simultaneous* work structure were also able to achieve comparable results for a higher cost but possibly lower latency. Additionally, we noticed that workers tend to make more changes to the narratives of the more popular match.

Work Style (HYB vs. CRO). It is clear from our experiments that for this task, the *crowd-only* work style outperforms its *hybrid* counterpart. We examined the outcomes from the strategies and found that while the narratives obtained using *hybrid*, work styles were sound and complete, they tended to be mechanical and repetitive whereas the narratives from a *crowd-only* work style were more creative. Indeed, creativity is something that algorithms are yet to learn.

Workforce Organization (IND vs. COL). Aside from the fact that it is easier to organize an *independent* workforce organization, we found in [30] that in writing a narrative for a topic such as soccer, which can be controversial and passionate, workers were more efficient with an *independent* workforce organization. We realized this when we examined the logs of exchanges between workers involved in the *collaborative* strategies and saw that they spent time arguing about the game statistics.

4.6 Discussion

4.6.1 Summary of Recommendations

In this study, we found that there is no single deployment strategy appropriate for the complex task of text creation. However, through our experiments, by considering different factors such as task type and task input characteristics, we drew up the following recommendations for requesters of translation, summarization, and narrative writing tasks.

For long text translation, we recommend, a hybrid work style combined with a sequential work structure. For shorter text, however, we recommend simultaneous work structure with either hybrid or crowd-only work styles. For summarization tasks, we recommend a crowd-only work style combined with a simultaneous work structure. Lastly, for narrative writing tasks that involve popular topics such as soccer, we recommend a sequential work structure combined with a hybrid work style.

4.6.2 Requesters-in-the-loop

Aside from recommendations to requesters, we advocate having requesters-in-the-loop. Since crowdsourcing itself can be considered as a creative task, we believe that having requesters-in-the-loop enhances the effectiveness of the process. For instance, in sequential strategies, requesters were involved in such a way that they could see intermediate results and decide whether to proceed or not. This intermediate evaluation helped requesters decide how many iterations were necessary. This has an immediate effect on the total task deployment cost and does not require that a requester spends the entire budget on a task. Involving requesters at different stages of task deployment contributes to making that process more transparent and provides intermediate feedback that could be useful in deciding or not to pursue a task. We believe that this shift to a requester-in-the-loop approach will be beneficial in increasing task throughput and could be integrated into a new strategy that helps requester set their expectations regarding cost, latency, and outcome quality.

4.7 Chapter Summary

In this chapter, we first defined a framework for the characterization and formalization of crowdsourcing deployment strategies. This framework characterizes strategies along three dimensions: *work structure*, *workforce organization* and *work style* and results in six distinct strategies. Each strategy is then adaptable through parameters such as the number of iterations or concurrent tasks deployed.

We then studied the effectiveness of these strategies when applied to text creation tasks and built a tool enabling simpler deployment of such tasks. Our results show that the particular characteristics of tasks, such as the need for creativity or subjectivity of the result, make a task respond differently to each strategy. We then drew recommendations for requesters wishing to crowdsource text creation task in regards to choosing a suitable strategy. These contributions are published in [32].

As future work, we may refine this analysis and consider the effect of different output evaluation schemes, budget amounts and workforce size on the effectiveness of these strategies.

Chapter 5

Personalized and Diverse Task Composition in Crowdsourcing

5.1 Background

Task assignment is a process in crowdsourcing wherein workers are matched to tasks and vice versa. Currently, there are two types of task assignment methods: push and pull. In *push methods*, the crowdsourcing platform assigns tasks to workers while in *pull methods*, workers find tasks via a user interface and self-appoint themselves to tasks [84]. Using *pull methods*, a worker needs to search for appropriate tasks from a list of tasks, which can be difficult especially when the list is huge. This can reduce overall throughput, accuracy, and engagement in a crowdsourcing system [87]. High search costs threaten to reduce the motivation to participate and cause workers to settle for less suitable tasks which can decrease the quality of the results [53]. Indeed, a thorough examination of TurkerNation[1], a forum for crowdworkers, revealed that workers spend non-negligible amounts of time discussing how to best select tasks depending on one's goals, which requesters to ban, and which skills are required for the latest tasks on AMT [98].

Efforts to address this include task recommendation frameworks that consider workers' preferences have been proposed. Ambati et al. modeled workers preferences based on a worker's profile, task metadata, and feedback from both workers and requesters and used a bag-of-word scheme to calculate similarities between

[1] *http://turkernation.com/*

tasks [21]. Yuen et al. based their model on a worker's work performance history and task searching history [129]. Lin et al. considered implicit signals about task preferences such as types of tasks that have been available and have been displayed, and the number of tasks workers select and complete [87]. Using these frameworks, available tasks are shown to workers.

Our research group addressed this issue by studying the problem of producing a personalized summary of tasks for each worker. We leveraged on the concept of Composite Items (CIs) and developed an approach that builds for each worker a set of valid Composite Tasks (CTs) and maximizes representativeness, diversity, and personalization. In this chapter, we will explain the concept of CTs and the experiments we performed to explore the impact of CTs on task throughput, worker retention, and outcome quality.

5.2 Composite Tasks

A *Composite Item* (CI) is a collection of items that satisfy a given constraint. It consists of a *central item* and a set of *satellite items* that are compatible with the central item [112]. For example, when creating a music playlist, a CI may be composed of a user's favorite song (central item) and similar music with similar genres (satellite items) by different artists and whose total duration is less than the desired play time. Leroy et. al formalized building representative CIs for heterogeneous items as an optimization problem and proposed a constraint-based fuzzy clustering algorithm that seamlessly integrates validity, cohesiveness, and representativity to solve it [85]. They applied their algorithm to three different data sets: Tourpedia[2], which contains a collection of heterogeneous points of interest in various European cities, MovieLens[3], a movie rating database, and BookCrossing[4], a data set that contains books and their user ratings.

Using crowdsourcing tasks as a data set, we can build a Composite Task (CT), which is a CI generated based on parameters specific to a worker.

[2] http://tour-pedia.org/

[3] https://grouplens.org/datasets/movielens/

[4] https://www.bookcrossing.com/

5.2.1 Preliminaries

To build composite tasks we consider the following elements.

- *Qualifications.* These represent an expertise domain that can be acquired by a worker or requested by a task.
- *Topics.* These describe both the content of tasks and workers' interests.
- *Worker.* A worker has the following attributes: worker ID, banned requesters or the list of requesters he does not wish to work for, qualifications, the number of tasks he usually wishes to complete in one session, and the topics of tasks he is interested in completing.
- *Task.* Each task has a unique identifier, qualifications required to perform a task, topics in which they fall in, the requester who created it, and the reward for completing the task.

Furthermore, we also consider the following similarity and diversity objectives.

- *Topic similarity* is measured either between tasks or between tasks and workers.
- *Requester similarity* is a boolean identifying whether two tasks are submitted by the same requester.
- *Reward similarity* is the normalized difference between the reward of two tasks.

Finally, we define our CT objectives: A CT should be *uniform* meaning it is similar in one dimension but *diverse* on a different dimension. For example, if a CT has similar rewards, it must have either diverse requesters or topics. It should be *personalized*, which means it should match the topics a worker is interested in, and it should be *representative* of the entire task set.

5.2.2 CT Creation

Based on an objective function that considers worker and task attributes, we used a fuzzy clustering algorithm that computes K clusters that are representative of the data set [19]. Fuzzy clustering was used as it allows each data point to participate in each cluster with a given weight. The algorithm computes the centroid of each cluster and the weights' distribution. In the context of CT creation, we can achieve representativeness by clustering the set of tasks into K clusters and selecting the tasks that constitute a CT in the vicinity of each centroid. To achieve uniformity, we can select tasks similar to the centroid. To achieve personalization, we can select tasks near the centroid position that match the worker's profile.

5.3 Experiments

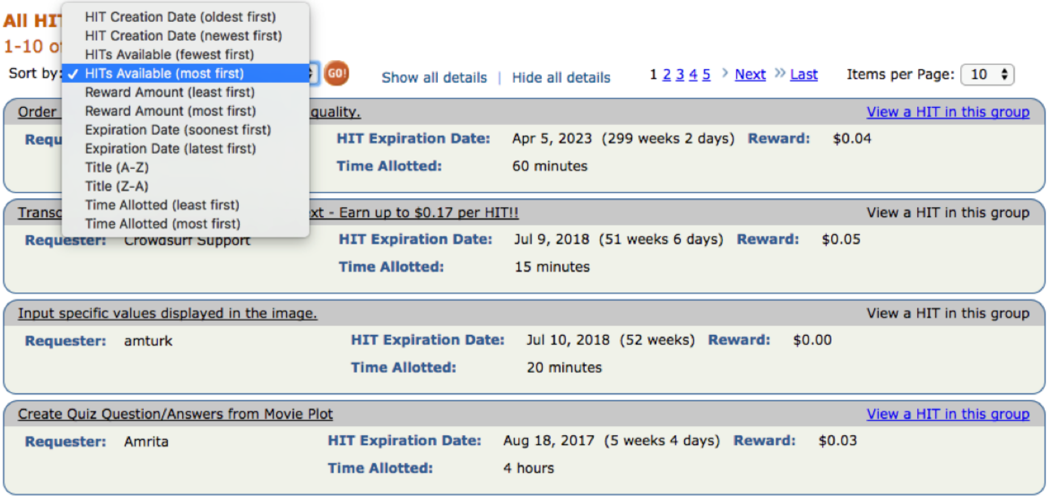
We evaluated the benefit of building CTs by performing offline experiments on personalization and diversity, and an online experiment on diversity. In the offline experiment on personalization, we first verified the utility of personalized CTs when compared to ranked lists through a user study. We then performed another user study to assess the impact of diverse CTs on workers' expected performance in the offline experiment on diversity. In the online experiment, we deployed tasks on a crowdsourcing platform testbed, presented them to workers using diverse CTs, and asked them to complete tasks as they see fit. We later measured task throughput, worker retention, and outcome quality.

5.3.1 Offline Experiments

5.3.1.1 Personalization

The goal of this experiment is to verify the utility of task composition by measuring the likelihood of workers selecting tasks presented to them. The tasks in a CT were clustered based on topic similarity, and they were not diversified to isolate the impact of personalization. In all cases, we eliminated tasks that the workers were not qualified for and tasks from banned requesters.

Tasks can be presented to workers in different ways, which are summarized in Table 5.1. In AMT, available tasks are presented to workers as a ranked list (Figure 5.1). This list can be sorted according to several fields such as creation date to find newest tasks, and reward to access highest-paying tasks. We refer to


All HI
1-10 o
Sort by:
HIT Creation Date (oldest first)
HIT Creation Date (newest first)
HITs Available (fewest first)
✓ HITs Available (most first)
Reward Amount (least first)
Reward Amount (most first)
Expiration Date (soonest first)
Expiration Date (latest first)
Title (A-Z)
Title (Z-A)
Time Allotted (least first)
Time Allotted (most first)
GO!
Show all details | Hide all details
1 2 3 4 5 › Next » Last
Items per Page: 10

Order ... quality. View a HIT in this group
Requ... HIT Expiration Date: Apr 5, 2023 (299 weeks 2 days) Reward: $0.04
Time Allotted: 60 minutes

Transc... xt - Earn up to $0.17 per HIT!! View a HIT in this group
Requester: Crowdsurf Support HIT Expiration Date: Jul 9, 2018 (51 weeks 6 days) Reward: $0.05
Time Allotted: 15 minutes

Input specific values displayed in the image. View a HIT in this group
Requester: amturk HIT Expiration Date: Jul 10, 2018 (52 weeks) Reward: $0.00
Time Allotted: 20 minutes

Create Quiz Question/Answers from Movie Plot View a HIT in this group
Requester: Amrita HIT Expiration Date: Aug 18, 2017 (5 weeks 4 days) Reward: $0.03
Time Allotted: 4 hours

FIGURE 5.1: Task Display in AMT

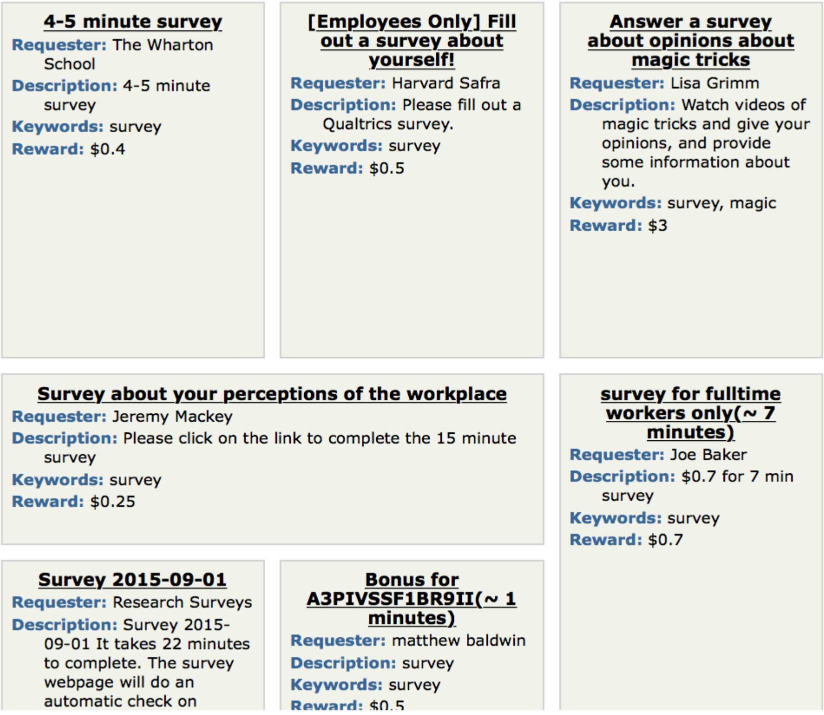
TABLE 5.1: Tasks display options

Task Display Options	**Description**
CRL	A list of tasks relevant to a worker and ranked by creation date
RRL	A list of tasks relevant to a worker and ranked by reward
SCT	A set of K CTs containing tasks of similar topics, not personalized, not diversified
PCT	A set of K CTs containing tasks of similar topics, personalized wrt. workers' topic preferences, not diversified

those sorting possibilities as *CRL* and *RRL* respectively. We set this ranked-list paradigm as the baseline in our experiments.

We designed a new task display option for CTs where tasks are no longer displayed as a flat list, but as groups of similar tasks (in terms of topic) personalized to match the worker's interests. Figure 5.2 shows a sample *CT* display option. Options *SCT* and *PCT* both build K CTs for workers. However, for *SCT*, we assumed that worker's qualifications are unknown as when a worker just started working in a platform. Hence, CTs are uniform, and representative, but not personalized to match the topics of the worker receiving them. In the case of *PCT*, we relied on the worker's qualifications to personalize the CTs and make them more appealing.

We set the number of CTs K to 6, and the fuzziness parameter m to 0.8. These settings correspond to moderate fuzziness and significant differences between CTs. We present more details of our experiment in the succeeding sections.



4-5 minute survey
Requester: The Wharton School
Description: 4-5 minute survey
Keywords: survey
Reward: $0.4

[Employees Only] Fill out a survey about yourself!
Requester: Harvard Safra
Description: Please fill out a Qualtrics survey.
Keywords: survey
Reward: $0.5

Answer a survey about opinions about magic tricks
Requester: Lisa Grimm
Description: Watch videos of magic tricks and give your opinions, and provide some information about you.
Keywords: survey, magic
Reward: $3

Survey about your perceptions of the workplace
Requester: Jeremy Mackey
Description: Please click on the link to complete the 15 minute survey
Keywords: survey
Reward: $0.25

survey for fulltime workers only(~ 7 minutes)
Requester: Joe Baker
Description: $0.7 for 7 min survey
Keywords: survey
Reward: $0.7

Survey 2015-09-01
Requester: Research Surveys
Description: Survey 2015-09-01 It takes 22 minutes to complete. The survey webpage will do an automatic check on

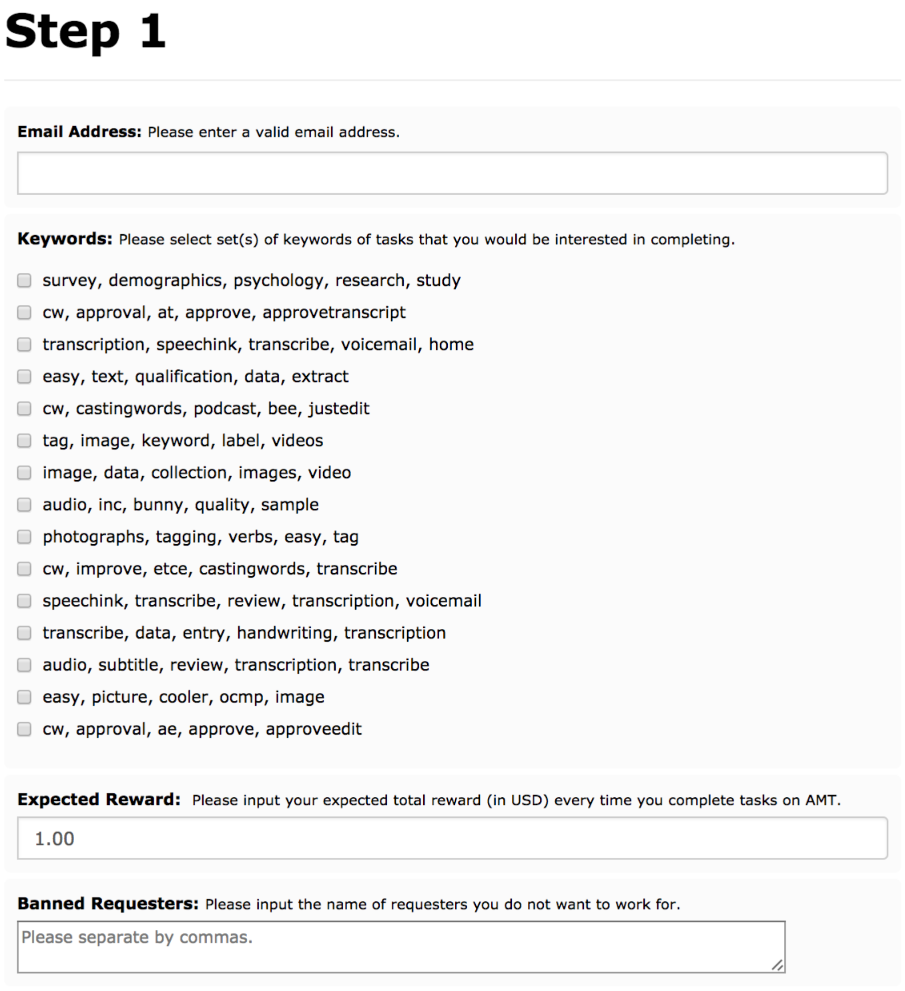
Bonus for A3PIVSSF1BR9II(~ 1 minutes)
Requester: matthew baldwin
Description: survey
Keywords: survey
Reward: $0.5

FIGURE 5.2: Personalized CT display

Task Dataset. We gathered a list of tasks available on AMT using a web crawler. From July 24 to August 12 in 2015, we retrieved the 300 most recent tasks and added them to a database. In practice, this means that we obtained almost all tasks submitted during the crawling period. We were able to collect a dataset of 25,644 individual tasks from 11,563 distinct projects. We recorded the tasks' *ID, creation date, title, description, keywords, requester name, reward, time allotted* and *qualifications.*

We then used this data set to generate different task displays for workers using the four different approaches described in Table 5.1. The Latent Dirichlet Allocation (LDA) [29] algorithm was applied on the keywords of tasks to discover 15 different topics of tasks. We described each topic as a bag of words, keeping the five most characterizing words. These topics are listed in Figure 5.3 (topic selection part). From the list of topics, we noticed the presence of tasks illustrating the typical variety of micro-tasks, with a prevalence of demographic surveys, image tagging tasks, and audio transcription for instance.

Worker Recruitment. We first posted a task on AMT to recruit workers. In this task, we explained to the workers that we were performing a 2-step study on task composition and asked them to indicate their desired reward, banned



Step 1

Email Address: Please enter a valid email address.

Keywords: Please select set(s) of keywords of tasks that you would be interested in completing.

□ survey, demographics, psychology, research, study
□ cw, approval, at, approve, approvetranscript
□ transcription, speechink, transcribe, voicemail, home
□ easy, text, qualification, data, extract
□ cw, castingwords, podcast, bee, justedit
□ tag, image, keyword, label, videos
□ image, data, collection, images, video
□ audio, inc, bunny, quality, sample
□ photographs, tagging, verbs, easy, tag
□ cw, improve, etce, castingwords, transcribe
□ speechink, transcribe, review, transcription, voicemail
□ transcribe, data, entry, handwriting, transcription
□ audio, subtitle, review, transcription, transcribe
□ easy, picture, cooler, ocmp, image
□ cw, approval, ae, approve, approveedit

Expected Reward: Please input your expected total reward (in USD) every time you complete tasks on AMT.

1.00

Banned Requesters: Please input the name of requesters you do not want to work for.

Please separate by commas.

FIGURE 5.3: Recruiting workers

requesters, and select their topics of interest from a checkbox list containing 15 options. The goal of this task is to build a profile for each worker participating in the user study.

We recruited a total of 70 workers in this study. The following are the five topics that are the most popular among workers recruited for this study:

- survey, demographics, psychology, research, study (95%)
- easy, picture, cooler, ocmp, image (70%)
- easy, text, qualification, data, extract (63%)
- tag, image, keyword, label, videos (61%)
- photographs, tagging, verbs, easy, tag (56%)

We can see that social studies were very popular, and so was annotating videos and images. Conversely, topics related to audio transcription had a low selection rate (15%). This was likely because such tasks require a quiet environment and dedicated equipment to listen to audio files. On average, each worker selected 6.2 topics, with a median at 5.

On average, workers indicated that they expected to earn \$1.23 each time they completed tasks on AMT. The standard deviation, however, was quite high, at \$1.24. Fifty percent of the workers indicated a value of \$1, with a minimum of \$0.10 and a maximum of \$10.

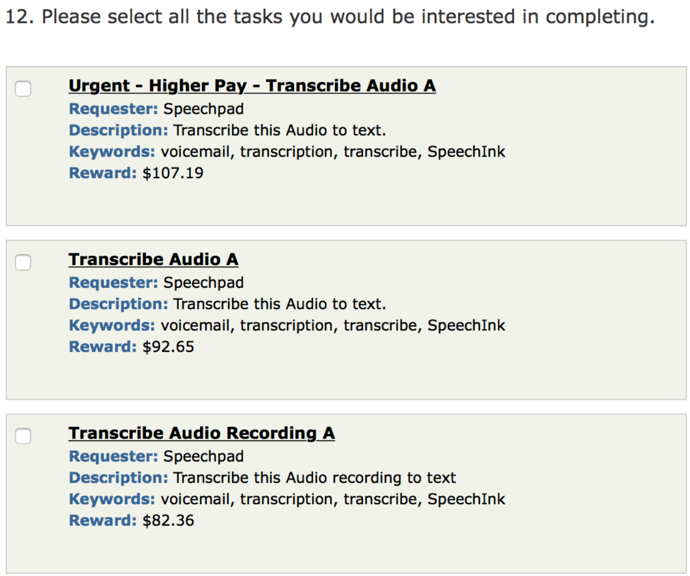
After the recruitment phase, each worker was paid \$0.10. The reward for this first task was low to encourage workers to perform the full study. However, they were informed that upon completion of all the steps in the study, they could receive a total of \$7.

Experiment Flow. For each worker, we built different task selections and displays according to the options described in Table 5.1. We first eliminated tasks the worker did not qualify for, and requesters the worker chose to ban. Then, we ranked tasks according to the creation date and reward to produce *CRL* and *RRL*. In the case of CTs, we took into account topic preferences from the worker's profile gathered in the recruitment step. Each CT was made of eight tasks. After generating the CTs and setting up the task display options, we invited the workers whose profiles we collected to do an evaluation task, where the workers were presented with the options generated from their profile.

At the end of the evaluation, the worker was given a reward of \$3.90 and a bonus of \$3, thus making the overall compensation for participating in the full study \$7.

Independent Evaluation. We performed an independent evaluation of the task display options, to assess the relevance of the tasks they contain (Figure 5.4).

We first evaluated the 4 task display options independently. Workers were presented with tasks, either as lists (*CRL* and *RRL*), or as a set of K=6 CTs (*SCT* and *PCT*) for each option. We asked them to select individual tasks that they would be interested in performing (Figure 5.4). We computed the average and median task acceptance rates for each option, which are given in Table 5.2.



12. Please select all the tasks you would be interested in completing.

☐ **Urgent - Higher Pay - Transcribe Audio A**
Requester: Speechpad
Description: Transcribe this Audio to text.
Keywords: voicemail, transcription, transcribe, SpeechInk
Reward: $107.19

☐ **Transcribe Audio A**
Requester: Speechpad
Description: Transcribe this Audio to text.
Keywords: voicemail, transcription, transcribe, SpeechInk
Reward: $92.65

☐ **Transcribe Audio Recording A**
Requester: Speechpad
Description: Transcribe this Audio recording to text
Keywords: voicemail, transcription, transcribe, SpeechInk
Reward: $82.36

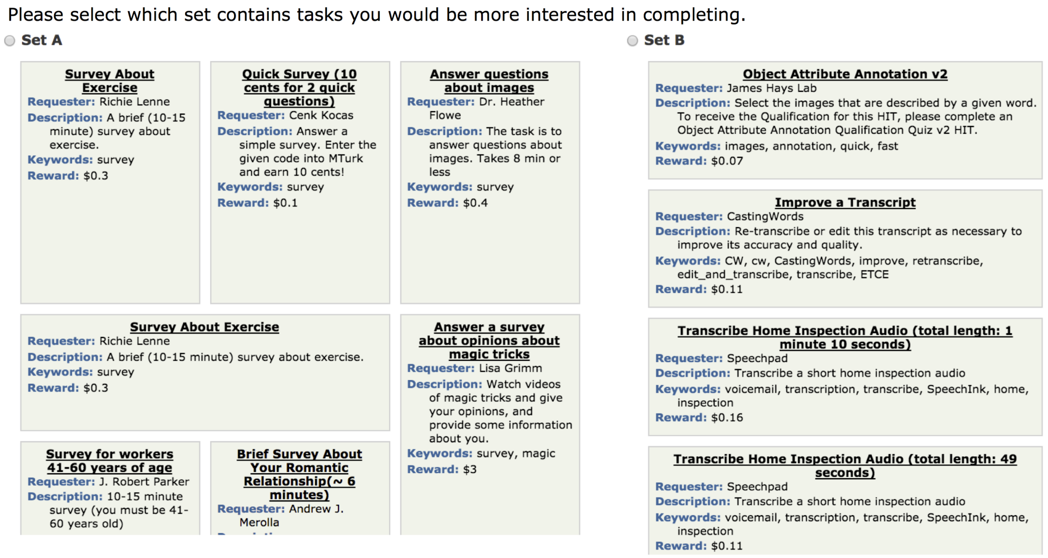
Figure 5.4: Independent evaluation task

CRL performed the worst, as listing tasks by creation date did not seem to bring any benefit to workers. We found that the workers were willing to perform an average of 30% of tasks, but we noted a very high variance. Indeed, some workers accepted most tasks and selected many of them regardless of the way they are displayed. However, selective workers, who constituted the majority of the workforce in this study, found few relevant tasks (median at 20%).

Sorting tasks by reward *RRL* proved to be an improvement, as it allowed workers to reach their reward objective more easily by selecting tasks with a high reward.

With the *SCT* option, task acceptance rate was 38% on average, with low variance. Despite not being personalized, the CTs generated by *SCT* had multiple advantages. Each CT was homogeneous as it contains tasks that were similar to each other. This allowed workers to quickly assess if they would be interested in a whole set of tasks (i.e. the CT), rather than having to independently evaluate individual tasks.

We also found that workers preferred performing several tasks of the same type in a row to be more efficient. Since each CT is representative of different topics of tasks, seeing a CT allows them to quickly get an overview the different type of tasks currently available on AMT. Furthermore, CTs present workers with sets of similar tasks that, taken simultaneously, let them complete a work session. Hence, workers do not have to skim through long lists of tasks to find suitable ones.



Please select which set contains tasks you would be more interested in completing.

○ **Set A**

Survey About Exercise
Requester: Richie Lenne
Description: A brief (10-15 minute) survey about exercise.
Keywords: survey
Reward: $0.3

Quick Survey (10 cents for 2 quick questions)
Requester: Cenk Kocas
Description: Answer a simple survey. Enter the given code into MTurk and earn 10 cents!
Keywords: survey
Reward: $0.1

Answer questions about images
Requester: Dr. Heather Flowe
Description: The task is to answer questions about images. Takes 8 min or less
Keywords: survey
Reward: $0.4

Survey About Exercise
Requester: Richie Lenne
Description: A brief (10-15 minute) survey about exercise.
Keywords: survey
Reward: $0.3

Answer a survey about opinions about magic tricks
Requester: Lisa Grimm
Description: Watch videos of magic tricks and give your opinions, and provide some information about you.
Keywords: survey, magic
Reward: $3

Survey for workers 41-60 years of age
Requester: J. Robert Parker
Description: 10-15 minute survey (you must be 41-60 years old)

Brief Survey About Your Romantic Relationship(~ 6 minutes)
Requester: Andrew J. Merolla

○ **Set B**

Object Attribute Annotation v2
Requester: James Hays Lab
Description: Select the images that are described by a given word. To receive the Qualification for this HIT, please complete an Object Attribute Annotation Qualification Quiz v2 HIT.
Keywords: images, annotation, quick, fast
Reward: $0.07

Improve a Transcript
Requester: CastingWords
Description: Re-transcribe or edit this transcript as necessary to improve its accuracy and quality.
Keywords: CW, cw, CastingWords, improve, retranscribe, edit_and_transcribe, transcribe, ETCE
Reward: $0.11

Transcribe Home Inspection Audio (total length: 1 minute 10 seconds)
Requester: Speechpad
Description: Transcribe a short home inspection audio
Keywords: voicemail, transcription, transcribe, SpeechInk, home, inspection
Reward: $0.16

Transcribe Home Inspection Audio (total length: 49 seconds)
Requester: Speechpad
Description: Transcribe a short home inspection audio
Keywords: voicemail, transcription, transcribe, SpeechInk, home, inspection
Reward: $0.11

FIGURE 5.5: Comparative evaluation task

TABLE 5.2: Independent evaluation: acceptance rate of tasks proposed

Task Display Option	**Median**	**Average**
CRL	20%	30%
RRL	27%	35%
SCT	35%	38%
PCT	48%	50%

PCT outperformed *SCT* by 12% on average. This confirms that accounting for worker topics preference when building composite tasks further improves the relevance of the results presented to the workers, and thus their satisfaction.

Comparative Evaluation. We then compared pairs of display options to understand the preferences of workers better. Workers were shown two task display options simultaneously, and they were asked which one they preferred (Figure 5.5). In this context, a display option was either the top-8 results of a ranked list (*RRL* and *CRL*), or one of the 6 CTs chosen at random (*PCT* and *SCT*). Hence, each display option presented eight tasks to the worker. The pairwise comparison results are given in Table 5.3.

We first compared the two list-based task display options and noticed that workers favor *RRL* over *CRL* 55% of the time. This result is consistent with the independent evaluation, which showed that workers are more likely to perform tasks displayed by *RRL*, since they have higher rewards.

Table 5.3: Comparative evaluation: pairwise preference of task display options

	CRL	**RRL**	**SCT**	**PCT**
CRL		45%	19%	21%
RRL	55%		14%	13%
SCT	81%	86%		44%
PCT	79%	87%	56%	

Table 5.4: Diversity and similarity configurations for CTs

Code	**Similarity Option**	**Diversity Option**
TN	Topic	None
TW	Topic	Reward
TR	Topic	Requester
WN	Reward	None
WT	Reward	Topic
WR	Reward	Requester

CT-based display options (*SCT* and *PCT*) consistently significantly outperformed list-based options (*RRL* and *CRL*). The ratio went from 79% for *PCT* against *CRL* to 87% for *PCT* against *RRL*. While the independent evaluation placed *SCT* relatively close to *RRL*, that was not the case in the comparative evaluation, as *SCT* was selected 86% of the time. This experiment showed a clear preference of workers for CT-based display options. Indeed, CTs allowed workers to browse coherent sets of similar tasks grouped together to allow workers to get a full work session. This facilitates the tedious operation of browsing and selecting tasks to perform. This experiment also confirmed that accounting for the worker's topic preferences when generating CTs is preferable, as *PCT* was selected over *SCT* 56% of the time.

5.3.1.2 Diversity

In the previous experiment, we verified that presenting workers personalized CTs improves their experience compared to ranking tasks by creation date or reward. In this experiment, we aim to evaluate the impact of diversity in producing CTs by measuring the likelihood of workers selecting the tasks presented to them through a user study.

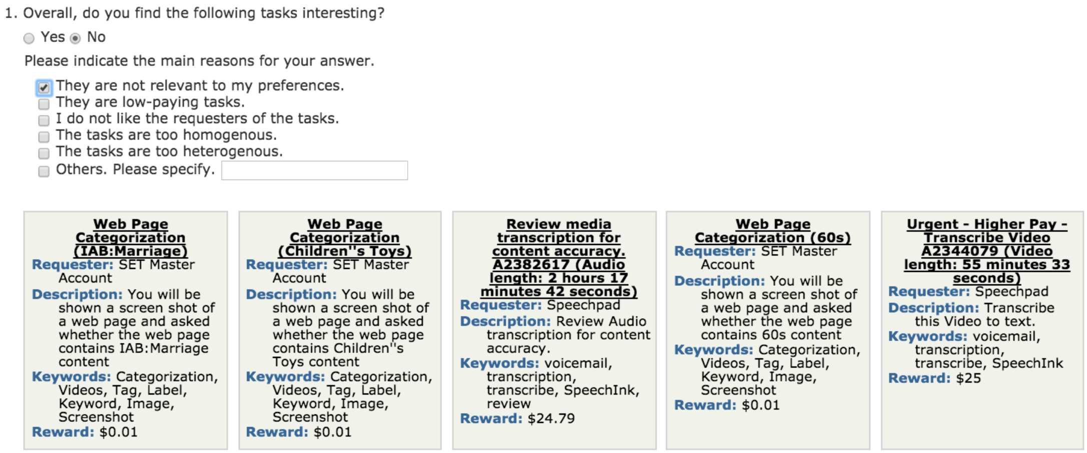
We first introduce different types of diversity in CTs. Table 5.4 summarizes the diversity options considered in our evaluation. CTs are generated by clustering tasks on topics or reward, and their diversity is enforced using topics, requesters, or rewards. This results in 6 options listed in the table. For example, TW consists of CTs containing tasks with similar topics and diverse rewards. We also consider the case where no diversity is enforced and referred to the resulting options as *TN* and *WN*. As discussed in Section 5.2.1, task diversity can be achieved using topics, requesters, or rewards. Topics typically describe the task type (e.g. survey) and the subject (e.g. psychology). Diversity in requesters may also result in topic diversity, as a requester typically posts tasks with similar topics. The reward can also promote diversity as similar tasks are rewarded similarly thus having diverse rewards may result in diverse tasks.

Task Dataset. We used the same dataset for the offline experiments in personalization. Additionally, we manually extracted the top 20 qualifications most required by the tasks by examining the task title, description, and associated keywords.

Worker Recruitment. We first collected worker profiles by posting a HIT that asks workers the following: the number of HITs they complete in one session, the topics of tasks they are interested in completing, the qualifications that have been assigned to them, and the name of the requesters they do not want to work for. We paid $0.05 for each profile collected.

Experiment Flow. For each worker, we built different CTs according to the options described in Table 5.4. We performed independent and comparative evaluations whereby workers were invited through AMT to evaluate one CT independently (see Figure 5.6 for the independent evaluation), and to compare two CTs (see Figure 5.7 for the comparative evaluation). Since there were six diversity options to evaluate independently and 15 pairs of options to be compared to each other, we assigned each worker to evaluate one CT and one pair. We had a total of 90 workers and recorded 15 evaluations for each diversity option and six evaluations for each pairwise comparison.

Independent Evaluation. For each combination of similarity and diversity, we asked workers if the CT was interesting, and what rating (5-star measure) it should get. The results of our independent evaluation are detailed in Table 5.5.



1. Overall, do you find the following tasks interesting?

○ Yes ● No

Please indicate the main reasons for your answer.

- ☑ They are not relevant to my preferences.
- □ They are low-paying tasks.
- □ I do not like the requesters of the tasks.
- □ The tasks are too homogenous.
- □ The tasks are too heterogenous.
- □ Others. Please specify.

Web Page Categorization (IAB:Marriage)
Requester: SET Master Account
Description: You will be shown a screen shot of a web page and asked whether the web page contains IAB:Marriage content
Keywords: Categorization, Videos, Tag, Label, Keyword, Image, Screenshot
Reward: $0.01

Web Page Categorization (Children''s Toys)
Requester: SET Master Account
Description: You will be shown a screen shot of a web page and asked whether the web page contains Children''s Toys content
Keywords: Categorization, Videos, Tag, Label, Keyword, Image, Screenshot
Reward: $0.01

Review media transcription for content accuracy. A2382617 (Audio length: 2 hours 17 minutes 42 seconds)
Requester: Speechpad
Description: Review Audio transcription for content accuracy.
Keywords: voicemail, transcription, transcribe, SpeechInk, review
Reward: $24.79

Web Page Categorization (60s)
Requester: SET Master Account
Description: You will be shown a screen shot of a web page and asked whether the web page contains 60s content
Keywords: Categorization, Videos, Tag, Label, Keyword, Image, Screenshot
Reward: $0.01

Urgent - Higher Pay - Transcribe Video A2344079 (Video length: 55 minutes 33 seconds)
Requester: Speechpad
Description: Transcribe this Video to text.
Keywords: voicemail, transcription, transcribe, SpeechInk
Reward: $25

FIGURE 5.6: Independent evaluation for diversity options

TABLE 5.5: Independent evaluation of diversity options

CT configuration	Interesting CTs	Avg. CT rating
TN	46.67%	2.80
TW	46.67%	3.23
TR	93.33%	3.60
WN	66.67%	3.37
WT	73.33%	3.68
WR	73.33%	3.67

The results clearly show that, given a similarity function, the CTs that workers liked the least are the ones that are were diversified (*TN* and *WN*). This affected both the proportion of CTs they consider interesting and their rating. While *TW* CTs were found as likely as *TN* CTs to be interesting, their average rating was significantly higher. This result confirms that diversity is indeed important for workers. More specifically, we can see that the *TR* configuration performed extremely well, with 93.33% CTs deemed interesting. These CTs are diversified by requester.

Our interpretation of these results is that workers like to perform tasks issued by different providers because it allows them to build a reputation and diversify their sources of income. Furthermore, having tasks that have similar topics but were submitted by different requesters offers little variety on the tasks performed, while keeping the overall focus in terms of topic. This seemed to be highly appreciated by workers. Similarly, workers preferred *WR* and *WT* over *WN*.

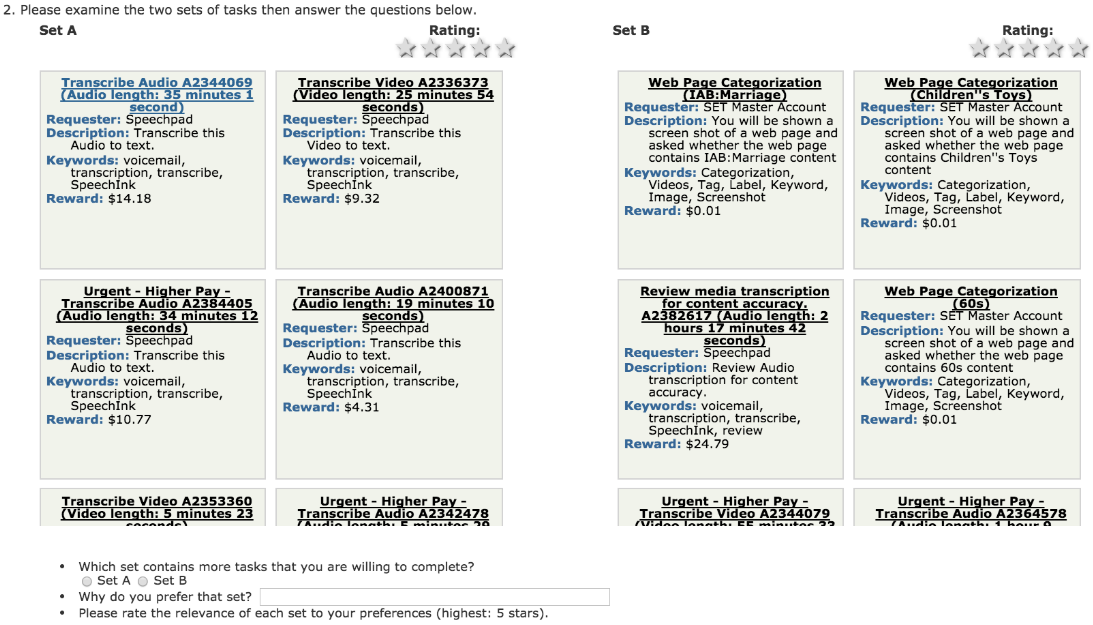
Workers seemed to have a preference for requester-based diversity (*TR* and *WR*). By examining the CTs generated by these options, we noticed that *WR* generated tasks that pay less than *TR*. This was a side effect of clustering by reward, as it eliminated some of the high-reward tasks which were less frequent. This was confirmed by workers after being asked explicitly why they like *TR* the most. *When workers are presented with homogeneous tasks with respect to their topics, they want to see tasks posted by different requesters.*

WR and *WT* performed similarly in terms of percentage of interesting CTs and average CT rating. However, feedback from workers on these configurations differed significantly. Workers commented that some of the tasks in the *WT* CTs were not relevant to their preferences. That may have been the case because the need to diversify by topic resulted in less relevance to workers. In the case of *WR*, workers commented that CTs tended to be low-paying, so they would be less likely to perform all the tasks of the CT. *From this feedback, it appears that workers are more willing to compromise on task relevance than on reward.*

We also observed that reward-based diversity (*TW*) performed poorly. By examining the data, we noticed that *TW* resulted in CTs containing highly homogeneous tasks in terms of their topics. In fact, the CTs generated by *TW* were very similar to those produced by *TN*. That could explain the lack of interest from workers in those CTs. When tasks were not diversified (*WN* and *TN*), workers preferred them to be clustered by reward. *That reinforces the observation that in the absence of diversity, workers want to use reward to choose tasks.*

Comparative Evaluation. We also performed a comparative evaluation where we showed two CTs generated using two different configurations and asked workers to select the CT they prefer and state the reason for their preference (Figure 5.7). Table 5.6 shows the result of the pairwise comparisons. For example, *TW* was preferred to *TN* 66.67% of the time. These results are consistent with the independent evaluation. *When tasks exhibit topic similarity, workers prefer to see reward diversity.*

WR and *TR* were the preferred options, with a small advantage for *WR*. *That reinforces the previous finding on workers' preference for requester-based diversity and topic clustering.* We also found that *TN* and *WN* have the lowest values, confirming that diversity, even if it is only a perception, always improves workers' satisfaction. Finally, *TW* also showed once again poor performance compared to other diversity options.



2. Please examine the two sets of tasks then answer the questions below.

Set A **Rating:** ☆☆☆☆☆

Transcribe Audio A2344069 (Audio length: 35 minutes 1 second)
Requester: Speechpad
Description: Transcribe this Audio to text.
Keywords: voicemail, transcription, transcribe, SpeechInk
Reward: $14.18

Transcribe Video A2336373 (Video length: 25 minutes 54 seconds)
Requester: Speechpad
Description: Transcribe this Video to text.
Keywords: voicemail, transcription, transcribe, SpeechInk
Reward: $9.32

Urgent - Higher Pay - Transcribe Audio A2384405 (Audio length: 34 minutes 12 seconds)
Requester: Speechpad
Description: Transcribe this Audio to text.
Keywords: voicemail, transcription, transcribe, SpeechInk
Reward: $10.77

Transcribe Audio A2400871 (Audio length: 19 minutes 10 seconds)
Requester: Speechpad
Description: Transcribe this Audio to text.
Keywords: voicemail, transcription, transcribe, SpeechInk
Reward: $4.31

Transcribe Video A2353360 (Video length: 5 minutes 23 seconds)

Urgent - Higher Pay - Transcribe Audio A2342478 (Audio length: 5 minutes 29

Set B **Rating:** ☆☆☆☆☆

Web Page Categorization (IAB:Marriage)
Requester: SET Master Account
Description: You will be shown a screen shot of a web page and asked whether the web page contains IAB:Marriage content
Keywords: Categorization, Videos, Tag, Label, Keyword, Image, Screenshot
Reward: $0.01

Web Page Categorization (Children''s Toys)
Requester: SET Master Account
Description: You will be shown a screen shot of a web page and asked whether the web page contains Children''s Toys content
Keywords: Categorization, Videos, Tag, Label, Keyword, Image, Screenshot
Reward: $0.01

Review media transcription for content accuracy. A2382617 (Audio length: 2 hours 17 minutes 42 seconds)
Requester: Speechpad
Description: Review Audio transcription for content accuracy.
Keywords: voicemail, transcription, transcribe, SpeechInk, review
Reward: $24.79

Web Page Categorization (60s)
Requester: SET Master Account
Description: You will be shown a screen shot of a web page and asked whether the web page contains 60s content
Keywords: Categorization, Videos, Tag, Label, Keyword, Image, Screenshot
Reward: $0.01

Urgent - Higher Pay - Transcribe Video A2344079 (Video length: 55 minutes 33

Urgent - Higher Pay - Transcribe Audio A2364578 (Audio length: 1 hour 9

- Which set contains more tasks that you are willing to complete?
  ○ Set A ○ Set B
- Why do you prefer that set?
- Please rate the relevance of each set to your preferences (highest: 5 stars).

Figure 5.7: Comparative evaluation for diversity options

Table 5.6: Pairwise comparison of CT configurations

	TN	*TW*	*TR*	*WN*	*WT*	*WR*
TN		33.33%	33.33%	83.33%	0.00%	33.33%
TW	66.67%		50.00%	50.00%	16.67%	33.33%
TR	66.67%	50.00%		50.00%	66.67%	33.33%
WN	16.67%	50.00%	50.00%		33.33%	33.33%
WT	100.00%	83.33%	33.33%	66.67%		50.00%
WR	66.67%	66.67%	66.67%	66.67%	50.00%	

5.3.2 Online Experiments

In the user study on diversity (offline experiment), we observed that workers preferred CTs diversified by topics and requesters (*TR*, *WR*, and *WT*). In this experiment, we aim to observe how CTs affect workers beyond their perception of the tasks. We performed online deployment wherein we asked workers to complete the tasks in CTs to evaluate the impact of diversity options on their performance. We let workers examine four CT configurations: *TR*, *WR*, *WT*, and *TN*. The first three were the ones preferred by workers in the user study described in the offline experiments in diversity, while *TN* served as a baseline.

5.3.2.1 Setup

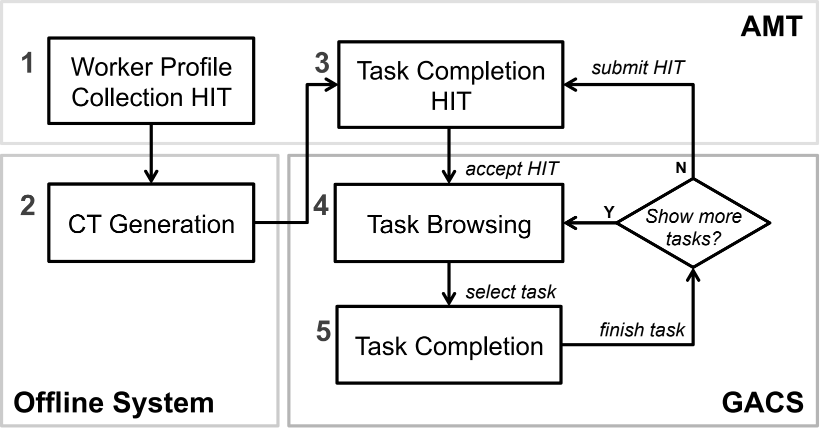


Figure 5.8: Flow of the online deployment

Task dataset. We used a dataset from CrowdFlower's open data library [6]. The dataset consists of 158,018 microtasks of 22 different task types such as tweet classification, searching information on the web, transcription of images, and sentiment analysis. Each task type was assigned a set of keywords that best describe its content, a reward that ranges from \$0.01 to \$0.12 proportional to its expected completion time, and a requester name selected at random from the AMT dataset.

Worker recruitment. We published a worker profile collection HIT in AMT as in the offline experiments then we assigned each worker a specific diversity option to examine. Since there were four diversity options, each option was examined by 30 workers. Five CTs consisting of 5 tasks each were then generated and uploaded to our platform GASC [107]. As illustrated in Figure 5.8, GASC is integrated with AMT for the recruitment and payment of workers. Once the CTs were uploaded, we invited workers to a task completion HIT in AMT that directed them to our platform where they were able to browse tasks and complete their preferred tasks. When a worker finishes all tasks available to her or simply wants to stop working, she submits all completed tasks in our platform, receives a unique code, which will serve her to receive her reward. We paid \$0.01 for the worker collection HIT and \$0.10 for the task completion HIT and paid as a bonus the workers' earnings from completing tasks in GASC.

5.3.2.2 Results

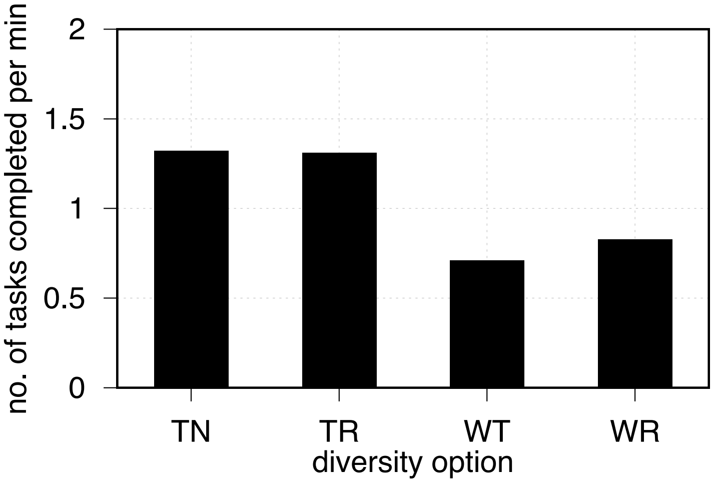


FIGURE 5.9: Task throughput

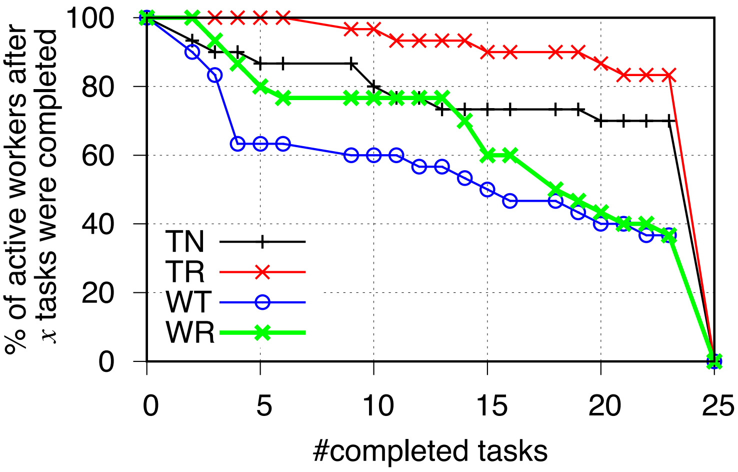


FIGURE 5.10: Worker retention

Task throughput and worker retention. Figure 5.9 shows results of task throughput. *Clearly, having tasks of similar topics significantly increases the performance of workers, as it allowed them to quickly complete the same kind of task several times in a row, with less context switching.* This also improves worker retention, as we can see in Figure 5.10 that workers stayed longer to complete tasks on the crowdsourcing platform for the *TN* and *TR* configurations. In the case of *TR*, 83% of workers performed all 25 tasks (5 CTs of 5 tasks) without leaving the system, which is higher than for *TN* (70%). This is particularly interesting, as requesters were assigned at random in this dataset. Hence, in practice, there was virtually no difference in terms of tasks selected between *TR* and *TN*. *However, the fact that workers were under the impression that they work for different requesters*

Table 5.7: Outcome quality evaluation

CT Configuration	**Total Tasks Completed**	**Quality**
TN	438	64.64%
TR	577	69.02%
WT	701	64.76%
WR	515	64.37%

may have encouraged them to perform more tasks. When using reward-based similarity, the majority of workers left before the end of the experiment, and only 37% remained after 25 tasks. We hence confirm our previous observations that *TR* offers the best combination of similarity and diversity, as *TR* obtains both the highest throughput and retention.

Outcome Quality. We manually built a ground-truth for 863 non-subjective tasks (e.g., opinion tasks were not chosen). We then compared workers' answers to those tasks to their ground-truths to assess crowdwork quality. The results are reported in Table 5.7. The quality of answers in different diversity configurations does not differ significantly except for *TR* CTs. Like *TN* CTs, *TR* CTs have similar tasks, which both have high throughput. However, the perceived diversity of requesters in *TR* encourages workers to maintain a high quality of work. *We conjecture that the exposure of workers to multiple requesters is a good incentive for higher quality standards.*

5.4 Discussion

In the user studies, we found that CTs significantly improved workers' experience, as it gave them direct access to a set of tasks that allowed them to meet their objective. Personalized CTs further improved workers' satisfaction as they reduced the time workers spent looking for relevant tasks. Additionally, we found that workers preferred CTs that exhibit some diversity, with similar topics and diverse requesters obtaining the highest approval from workers. We conjecture that it is because focusing on tasks having the same topic is more efficient while performing tasks for different requesters allows them to build a reputation and diversify their sources of income. We also found that workers were more willing to compromise on task relevance than on reward. That reinforces the observation that in the absence of diversity, workers consider the reward in choosing tasks. That is compatible with our previous finding showing that in the absence of CTs, workers prefer to see tasks ranked by reward.

In the online deployment, we found that having tasks with similar topics significantly increased the performance of workers, as it allowed them to quickly complete the same kind of task multiple times, with less context switching. This led to higher task throughput. We also observed that this efficiency encouraged workers to stay longer and complete more tasks. Worker retention was maximized with a combination of topic similarity and requester diversity. This configuration was also the one that obtained the highest crowdwork quality.

5.5 Chapter Summary

We presented task composition in crowdsourcing and evaluated the effects of personalization and diversity in forming tasks. A thorough examination of TurkerNation[5] revealed that workers spend non-negligible amounts of time discussing how to best select tasks depending on one's goals, which requesters to ban, and which skills are required for the latest tasks on AMT. Therefore, we proposed to provide to workers Composite Tasks (CTs), that match their profiles and exhibit diversity. Our extensive experiments validated the assumption that workers prefer personalized CTs to their non-personalized counterparts and that CTs are superior to ranked lists of tasks. We then deployed tasks and observed workers completing them. This deployment resulted in some observations that can serve as a basis for future research in crowdsourcing. Indeed, while our observations are empirical, we measured some performance indicators that lead to a finer understanding of the workforce. Task throughput is higher with similar tasks and worker retention even better with tasks that exhibit similar keywords but that are proposed by different requesters. Moreover, when tasks are similar, workers prefer to see diversity in requesters or reward. Finally, crowdwork quality is highest in the case workers stay longer in the system and become proficient at completing similar tasks proposed by different requesters. Consequently, workers care about their image and strategically expose their work to multiple requesters. This work is also reported in [19].

[5] http://turkernation.com/

Chapter 6

Conclusion

6.1 Summary of Contributions

In this research, I approached quality management in crowdsourcing based on the sub-processes involved, specifically: task design, task deployment, and task assignment. I first experimented on factors affecting task design. I found no significant difference in the quality achieved from simple and more complex versions of a data extraction task and observed that the performance of paid and unpaid workers are comparable in a sentiment analysis task.

I, along with my collaborators, then proposed deployment strategies along three dimensions: work structure, workforce organization, and work style, and developed a tool to enable simpler deployment in a crowdsourcing platform. We then studied the effectiveness of the strategies when applied to text creation tasks and drafted recommendations for both crowdsourcing researchers and practitioners.

Lastly, for task assignment, I, again together with my collaborators, validated a fuzzy clustering-based method for building composite tasks or a personalized summary of tasks for crowd workers. We found that personalization improves the workers' overall experience and that diversifying tasks can improve the workers' output quality.

6.2 Applicability of Crowdsourcing to Tasks

Aside from studying quality control in crowdsourcing, I was also able to study crowdsourcing's applicability to different types of tasks namely data extraction, sentiment analysis, language translation, text summarization, and narrative writing. These tasks were chosen because they are intuitively easier for humans than algorithms yet there are numerous studies on automatic solutions. These tasks can be classified into two: microtasks and macrotasks. Microtasks are tasks that require minimal time and cognitive effort but when combined can result in major accomplishments [101]. They are typically simple, repetitive, independent, and short. Macrotasks are more complex tasks that are context-heavy, interdependent, require more cognitive effort, and may take many hours to complete [57]. In the tasks used in this research, data extraction and sentiment analysis can be considered as microtasks whereas language translation, text summarization, and narrative writing can be considered as macrotasks.

Currently, the application of crowdsourcing to sentiment analysis and data extraction is widely used. In fact, popular crowdsourcing platforms such as AMT and CrowdFlower have project templates and wizards to enable requesters to design and deploy these types of tasks easily. The platforms also provide quality control measures for these tasks such as automatic answer aggregation for sentiment analysis tasks in CrowdFlower using workers' trust scores and implementing qualification tests in AMT. However, the requester is still the main quality control manager. As a requester, I believe that for microtasks, increasing the number of evaluators per task may be enough to improve the quality of results, at the expense of cost.

Language translation is also a task that can be commonly found in crowdsourcing marketplaces. However, the text to be translated is typically short (one sentence or less). Since the text in the experiments in Chapter 4 were much longer, many answers from the crowd turned out to be outputs of machine translators such as Google Translate. Thus I believe that the strategies proposed in Chapter 4 can be useful when translating longer text.

Although automatic tools also exist for summarization and narrative writing tasks, they are less popular and not as readily available to the public as Google Translate. Thus, workers seemed to be more motivated to write text from scratch. However, in cases when they were asked to improve a given text, they tended to check the spelling and grammar rather than the actual content of the narrative or summary. To avoid this in the future, we can give more precise instructions to the workers on how to perform improvement tasks.

More improvements can be done in the experiments performed, but I believe crowdsourced solutions are appropriate for the tasks considered in this research.

6.3 Insights

Based on my experience in conducting experiments, I recommend users to consider the following when designing their crowdsourcing projects: task assessment, experimentation, and human factors.

Task Assessment. As I learned how complex the crowdsourcing process could be, requesters must first assess the applicability of crowdsourcing to his project. Ideally, requesters use crowdsourcing under the assumption that it is more effective in terms of quality, cost, and latency than other options (knowledge, skills, and expertise available in-house or automatic methods) [118]. Otherwise, the benefits of crowdsourcing may not be taken advantage of. For example, in the data extraction task, the benefit of crowdsourcing over the manual method might not be obvious because there were only 69 entries to extract data from. However, that is because only a sample set was used for these experiments. Currently, there are hundreds of entries in the digital archive that would be tedious to do manually thus I believe that for such task, crowdsourcing is a reasonable approach.

Experimentation. Secondly, I recommend requesters to test their task designs periodically using real workers. While a good project design backed by theories and best practices is essential to the success of any project, there may be factors that could only be discovered upon actual project deployment. For example, when our team first deployed improvement tasks, we did not employ any intermediate evaluation techniques. However, after running several tasks, we realized that some workers submit the original text without any improvements. Thus we decided to add an intermediate evaluation wherein we check the difference between the original text and the submitted text.

Human Factors. Lastly, I urge requesters to think of the workers when they are designing the tasks. Since the ones who will complete the tasks are humans and not machines, it is important to consider the workers' motivations and behavior. Moreover, understanding human factors affects the performance of workers [22], which in turn affects the quality of crowdsourced work. In the sentiment analysis experiment, where unpaid workers were employed, it was necessary to ensure that the tasks were easy to complete as not to tax the volunteers. Thus buttons were used as seen in Figure 3.2 so that workers only needed one click to provide an answer.

6.4 Future Directions

Currently, crowdsourcing can be perceived as a creative process, an art rather than science. The goal of researchers is to make the processes more scientific. As a relatively new research field, many challenges need to be addressed. Molina et al. describe challenges for future work in data crowdsourcing from interface design and testing, platforms, fluidity in marketplaces, task difficulty, batching of tasks, holistic process optimization to managing worker identities, boredom, laziness, experiences, biases, and incentives [52]. They also discuss how prior information can be incorporated in task results and suggest integration of crowdsourcing with active learning. Kittur et. al also point out future directions in crowd work processes, crowd computation and crowd workers based on information provided by crowd workers and theory from organizational behavior and distributed computing [72].

In this research, I believe that the main challenge that must be addressed is ensuring the validity of crowdsourced experiments. Currently, empirical data has been published. However, in the future, it is necessary to publish statistically significant data. To start with, SurveyMonkey's[1] sample size calculator [1] can be adopted for crowdsourced user studies. However, for experiments such as in deployment strategies, further research is necessary on the appropriate number of iterations, crowd size, and the number of trials.

Another direction which has been slightly explored by our research team is fairness and transparency in crowdsourcing, which are two key issues that are of

[1] https://www.surveymonkey.com/

interest today in ethics[2]. Research on fairness has primarily focused on studying worker compensation or on helping requesters identify malevolent workers. To promote transparency, tools and plug-ins have been developed to disclose computed information such as workers' performance and requesters' rating. However, existing works are fragmented, and we believe that a holistic approach to both fairness and transparency is necessary because of the dependencies between crowdsourcing processes. In [33], we propose to develop fairness check benchmarks and algorithms for existing crowdsourcing systems and a declarative high-level language to specify fairness rules, in the future.

6.5 Final Remarks

In this research, I studied the phenomenon of crowdsourcing and contributed to quality control techniques based on the sub-processes involved in crowdsourcing. I believe the findings in the experiments can be used as data sets that quality control recommender systems can learn from. From the experiments performed in this research, I gained a better understanding of the difficulty and complexity of the crowdsourcing process which I can use in the future to further advance crowdsourcing quality control research.

[2] http://fatml.org/

Appendix A

CDeployer User's Manual

CDeployer is a tool developed in Java to aid the implementation of the deployment strategies proposed in Chapter 4. Details regarding its usage and functionalities are described in this appendix.

A.1 Requirements

- **Amazon Web Services (AWS) Account**. Since *CDeployer* uses the AMT API, it is necessary to have an account. In addition, AWS [1] keys are necessary. The following must be written in the *$HOME_ DIRECTORY/.aws/credentials* file.

```
[default]
aws_access_key_id=[aws_access_key_id]
aws_secret_access_key=[aws_secret_access_key]
```

- **Main Configuration File**. This file must be named *cDeployer.properties.* It instructs *CDeployer* where to get the input files and send notification email. It also specifies whether or not to use the AMT Sandbox[2].

```
basedir=./
notificationEmail=ria@db.ics.keio.ac.jp
sandbox=false
```

[1] https://aws.amazon.com/

[2] http://requestersandbox.mturk.com

- **CDeployer JAR file**. This can be obtained by compiling the sources published in GitHub[3]. In the succeeding sections, we refer to the generated jar file as *cdeployer.jar*.

A.2 Functionalities

The following actions can be performed using *CDeployer*.

1. **Create HIT** - The requester can create the following types of HITs described in Section 4.5.1.2: *Creation HIT* and *Improvement HIT*. To create a new HIT, the requester must prepare the following:

   - *HIT configuration File* - this contains the metadata about the HIT such as title, description, keywords, reward, and duration.
   - *HTML template* - this contains the user interface for the task.
   - *Additional Input (optional)* - this can be pasted to the HTML template, in case the requester has multiple text inputs.

   HIT ids of all launched HITs are stored in a local log file. The usage is as in the following.

   ```
   $ java -jar cdeployer.jar -a Create
   usage: CreateHIT
   -a,--action <arg>      Should be Create
   -c,--config <arg>      Path to the configuration file
   -tp, --template <arg>  HTML template
   -tf,--textfile <arg>   Space separated list of HTML files to populate
   the HIT
   ```

2. **Review Results** - The answer is compared to a benchmark. If the degree of similarity is within the given threshold, the answer is rejected. Otherwise, the answer is accepted and the worker can be assigned a qualification when necessary. The requester must input the HIT ID to review, text to compare the answers to, qualification to assign. The usage is as in the following.

   ```
   $ java -jar cdeployer.jar -a Review
   usage: ReviewHIT
    -a,--action <arg>      Should be Review
   ```

[3] https://github.com/riamaehb/CDeployer

```
 -b,--benchmark <arg>   Space separated list of files where the
benchmark
                        texts are stored
 -h,--hitID <arg>       ID of the HIT to review
 -q,--qualifs <arg>     Space separated list of qualification IDs to
                        assign to workers
 -t,--threshold <arg>   Threshold used to compare contributions to the
                        benchmark
```

3. **Run Eval HIT** - This allows the requester to create the evaluation HIT described in Section 4.5.1.2. The usage is as in the following.

```
$ java -jar cdeployer.jar -a Eval
usage: EvalHIT
 -a,--action <arg>      Should be Eval
 -c,--config <arg>      Path to the configuration file
 -h,--hitID <arg>       ID of the HIT to review
 -tp, --template <arg>  HTML template
 -tf,--textfile <arg>   Space separated list of text files to populate
the
                        HIT
```

A.3 Other Features

- Email Notifications. An email is sent to the requester when a task is completed.
- Logging. All operations ran are logged in local files.

A.4 Example

In this example, we will implement a SEQ-IND-CRO for a narrative writing task. To execute this, we need one to run the following: Create HIT (creation HIT), Create HIT (improvement HIT), and Review HIT.

1. Create HIT (creation HIT)
    - HIT Configuration File (*NW-create.properties*)
    - HTML Template (NW-create.html)

- Additional input (NW-input.html)
- Command

```
    java -jar cdeployer.jar -a Create -c NW.properties -tp NW-
create.html -tf NW-input.html -tt narrative
```

2. Create HIT (improvement HIT) - Ask the worker to improve the output of the first HIT.

   - HIT Configuration File (*NW-improve.properties*)
   - HTML Template (NW-improve.html)
   - Additional input (NW-ans1.html) - This is the text output of the creation HIT.

   ```
       java -jar cdeployer.jar -a Create -c NW.properties -tp NW-
   improve.html -tf NW-ans1.html -tt narrativeimprove
   ```

3. Review HIT - Check if the worker improved the previous text.

   - Benchmark (NW-ans1.html)
   - HIT ID - [hitID] which can be obtained from the logs
   - Threshold - 0.9. This tells CDeployer to reject the answer if the texts are 0.95% similar.
   - Qualification (optional) - [qualifID] can be given to worker whose answers get accepted

   ```
       java -jar cdeployer.jar -a Review -b NW-ans1.html -h [hitID] -q
    [qualifID] -t 0.9
   ```

Bibliography

[1] Calculating the number of respondents you need. https://help.surveymonkey.com/articles/en_US/kb/How-many-respondents-do-I-need. [Online; accessed 01-Jul-2017].

[2] Crowdgrader: Peer grading for your classroom. http://www.crowdgrader.org. [Online; accessed 15-Apr-2017].

[3] Crowdmed, online medical diagnosis, differential diagnosis. https://www.crowdmed.com/. [Online; accessed 15-Apr-2017].

[4] Crowdsourcing: A definition. http://crowdsourcing.typepad.com/cs/2006/06/crowdsourcing_a.html. [Online; accessed 30-Mar-2016].

[5] crowdsourcing, n. http://www.oed.com/view/Entry/376403?redirectedFrom=crowdsourcing. [Online; accessed 30-Mar-2016].

[6] Data for everyone library | crowdflower. https://www.crowdflower.com/data-for-everyone/. [Online; accessed 15-Apr-2017].

[7] Free community-based mapping, traffic & navigation app. http://www.waze.com. [Online; accessed 30-Mar-2016].

[8] Home | globalxplorer. https://www.globalxplorer.org/. [Online; accessed 15-Apr-2017].

[9] How to calculate a confidence score. https://success.crowdflower.com/hc/en-us/articles/201855939-How-to-Calculate-a-Confidence-Score. [Online; accessed 01-Jul-2017].

[10] Human computation. http://www.hcjournal.org. [Online; accessed 30-Mar-2016].

[11] James 'jim' nicholas gray. http://amturing.acm.org/award_winners/gray_3649936.cfm. [Online; accessed 30-Mar-2016].

[12] Teaching evaluation comments, 2006-2012, dr. jonathan cox, department of mathematical sciences, state university of new york at fredonia. http://www.fredonia.edu/faculty/math/JonathanCox/TeachEvcommentsWeb.doc, 2012. [Online; accessed 31-Mar-2015].

[13] The ultimate crowdsourcing framework - pybossa. http://pybossa.com/, 2017. [Online; accessed 01-Apr-2017].

[14] Eugene Agichtein, Carlos Castillo, Debora Donato, Aristides Gionis, and Gilad Mishne. Finding high-quality content in social media. In *Proceedings of the 2008 international conference on web search and data mining*, pages 183–194. ACM, 2008.

[15] Takako Aikawa, Kentaro Yamamoto, and Hitoshi Isahara. The impact of crowdsourcing post-editing with the collaborative translation framework. In *Advances in Natural Language Processing*, pages 1–10. Springer, 2012.

[16] Ahmet Aker, Mahmoud El-Haj, M-Dyaa Albakour, Udo Kruschwitz, et al. Assessing crowdsourcing quality through objective tasks. In *LREC*, pages 1456–1461. Citeseer, 2012.

[17] Samir Aknine, Suzanne Pinson, and Melvin F Shakun. A multi-agent coalition formation method based on preference models. *Group Decision and Negotiation*, 13(6):513–538, 2004.

[18] Mohammad Allahbakhsh, Boualem Benatallah, Aleksandar Ignjatovic, Hamid Reza Motahari-Nezhad, Elisa Bertino, and Schahram Dustdar. Quality control in crowdsourcing systems: Issues and directions. *IEEE Internet Computing*, 17(2):76–81, 2013.

[19] Maha Alsayasneh, Sihem Amer-Yahia, Eric Gaussier, Vincent Leroy, Julien Pilourdault, Ria Mae Borromeo, Motomichi Toyama, and Jean-Michel Renders. Personalized and diverse task composition in crowdsourcing. *Knowledge and Data Engineering, IEEE Transactions on*, 2017. under minor revision.

[20] Vamshi Ambati, Stephan Vogel, and Jaime Carbonell. Collaborative workflow for crowdsourcing translation. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*, pages 1191–1194. ACM, 2012.

[21] Vamshi Ambati, Stephan Vogel, and Jaime G Carbonell. Towards task recommendation in micro-task markets. *Human computation*, 11:11, 2011.

[22] Sihem Amer-Yahia and Senjuti Basu Roy. Toward worker-centric crowdsourcing. *IEEE Data Eng. Bull.*, 39(4):3–13, 2016.

[23] Dimitra Anastasiou and Rajat Gupta. Comparison of crowdsourcing translation with machine translation. *Journal of Information Science*, 37(6):637–659, 2011.

[24] Paul André, Robert E Kraut, and Aniket Kittur. Effects of simultaneous and sequential work structures on distributed collaborative interdependent tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 139–148. ACM, 2014.

[25] Daniel Archambault, Tobias Hoßfeld, and Helen C Purchase. Crowdsourcing and human-centred experiments (dagstuhl seminar 15481). In *Dagstuhl Reports*, volume 5. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2016.

[26] Daniel W Barowy, Charlie Curtsinger, Emery D Berger, and Andrew McGregor. Automan: A platform for integrating human-based and digital computation. *Acm Sigplan Notices*, 47(10):639–654, 2012.

[27] Michael S Bernstein, Joel Brandt, Robert C Miller, and David R Karger. Crowds in two seconds: Enabling realtime crowd-powered interfaces. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 33–42. ACM, 2011.

[28] Jeffrey P. Bigham, Chandrika Jayant, Hanjie Ji, Greg Little, Andrew Miller, Robert C. Miller, Robin Miller, Aubrey Tatarowicz, Brandyn White, Samuel White, and Tom Yeh. Vizwiz: nearly real-time answers to visual questions. In *UIST*, pages 333–342, 2010.

[29] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[30] Ria Mae Borromeo, Maha Alsaysneh, Sihem Amer-Yahia, and Vincent Leroy. Crowdsourcing strategies for text creation tasks. In *Proceedings of the 20th International Conference on Extending Database Technology, EDBT 2017, Venice, Italy, March 21-24, 2017.*, pages 450–453, 2017.

[31] Ria Mae Borromeo, Thomas Laurent, and Motomichi Toyama. The influence of crowd type and task complexity on crowdsourced work quality. In *Proceedings of the 20th International Database Engineering & Applications Symposium*, pages 70–76. ACM, 2016.

[32] Ria Mae Borromeo, Thomas Laurent, Motomichi Toyama, Maha Alsayasneh, Sihem Amer-Yahia, and Vincent Leroy. Deployment strategies for crowdsourcing text creation. *Information Systems*, 71:103–110, 2017.

[33] Ria Mae Borromeo, Thomas Laurent, Motomichi Toyama, and Sihem Amer-Yahia. Fairness and transparency in crowdsourcing. In *EDBT*, pages 466–469, 2017.

[34] Ria Mae Borromeo and Motomichi Toyama. Automatic vs. crowdsourced sentiment analysis. In *Proceedings of the 19th International Database Engineering & Applications Symposium*, pages 90–95. ACM, 2015.

[35] Ria Mae Borromeo and Motomichi Toyama. An investigation of unpaid crowdsourcing. *Human-centric Computing and Information Sciences*, 6(1):11, 2016.

[36] Irma Borst. *Understanding Crowdsourcing: Effects of motivation and rewards on participation and performance in voluntary online activities*. Number EPS-2010-221-LIS. 2010.

[37] Ioannis Boutsis and Vana Kalogeraki. Crowdsourcing under real-time constraints. In *Parallel & Distributed Processing (IPDPS), 2013 IEEE 27th International Symposium on*, pages 753–764. IEEE, 2013.

[38] Jonathan Bragg, Andrey Kolobov, Mausam Mausam, and Daniel S Weld. Parallel task routing for crowdsourcing. In *Second AAAI Conference on Human Computation and Crowdsourcing*, 2014.

[39] Chris Callison-Burch. Fast, cheap, and creative: evaluating translation quality using amazon's mechanical turk. In *Proceedings of the 2009 Conference*

on Empirical Methods in Natural Language Processing: Volume 1-Volume 1, pages 286–295. Association for Computational Linguistics, 2009.

[40] Donald J Campbell. Task complexity: A review and analysis. *Academy of management review*, 13(1):40–52, 1988.

[41] Miguel Camus. Ph world's no. 1 in terms of time spent on social media. `http://technology.inquirer.net/58090/ph-worlds-no-1-terms-time-spent-social-media`. [Online; accessed 15-Apr-2017].

[42] Lydia B Chilton, Greg Little, Darren Edge, Daniel S Weld, and James A Landay. Cascade: Crowdsourcing taxonomy creation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1999–2008. ACM, 2013.

[43] Edward Deci and Richard M Ryan. Intrinsic motivation and self-determination in human behavior. *Perspectives in Social Psychology*, 1985.

[44] Anhai Doan, Raghu Ramakrishnan, and Alon Y Halevy. Crowdsourcing systems on the world-wide web. *Communications of the ACM*, 54(4):86–96, 2011.

[45] Steven Dow, Anand Kulkarni, Scott Klemmer, and Björn Hartmann. Shepherding the crowd yields better work. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*, pages 1013–1022. ACM, 2012.

[46] Julie S Downs, Mandy B Holbrook, Steve Sheng, and Lorrie Faith Cranor. Are your participants gaming the system?: screening mechanical turk workers. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2399–2402. ACM, 2010.

[47] E Estellés-Arolas and F González-Ladrón-de Guevara. Towards an integrated crowdsourcing definition. *Journal of Information science*, 38(2):189–200, 2012.

[48] Ju Fan, Guoliang Li, Beng Chin Ooi, Kian-lee Tan, and Jianhua Feng. icrowd: An adaptive crowdsourcing framework. In *SIGMOD*, pages 1015–1030, 2015.

[49] Ju Fan, Meiyu Lu, Beng Chin Ooi, Wang-Chiew Tan, and Meihui Zhang. A hybrid machine-crowdsourcing system for matching web tables. In *Data Engineering (ICDE), 2014 IEEE 30th International Conference on*, pages 976–987. IEEE, 2014.

[50] Ailbhe Finnerty, Pavel Kucherbaev, Stefano Tranquillini, and Gregorio Convertino. Keep it simple: Reward and task design in crowdsourcing. In *Proceedings of the Biannual Conference of the Italian Chapter of SIGCHI*, page 14. ACM, 2013.

[51] Michael J Franklin, Donald Kossmann, Tim Kraska, Sukriti Ramesh, and Reynold Xin. Crowddb: answering queries with crowdsourcing. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*, pages 61–72. ACM, 2011.

[52] Hector Garcia-Molina, Manas Joglekar, Adam Marcus, Aditya Parameswaran, and Vasilis Verroios. Challenges in data crowdsourcing. *IEEE Transactions on Knowledge and Data Engineering*, 28(4):901–911, 2016.

[53] David Geiger and Martin Schader. Personalized task recommendation in crowdsourcing information systems—current state of the art. *Decision Support Systems*, 65:3–16, 2014.

[54] Jade Goldstein, Vibhu Mittal, Jaime Carbonell, and Mark Kantrowitz. Multi-document summarization by sentence extraction. In *Proceedings of the 2000 NAACL-ANLPWorkshop on Automatic summarization-Volume 4*, pages 40–48. Association for Computational Linguistics, 2000.

[55] Jorge Goncalves, Denzil Ferreira, Simo Hosio, Yong Liu, Jakob Rogstadius, Hannu Kukka, and Vassilis Kostakos. Crowdsourcing on the spot: altruistic use of public displays, feasibility, performance, and behaviours. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pages 753–762. ACM, 2013.

[56] Shinsuke Goto, Donghui Lin, and Toru Ishida. Crowdsourcing for evaluating machine translation quality. In *LREC*, pages 3456–3463, 2014.

[57] Daniel Haas, Jason Ansel, Lydia Gu, and Adam Marcus. Argonaut: macrotask crowdsourcing for complex data processing. *Proceedings of the VLDB Endowment*, 8(12):1642–1653, 2015.

[58] Harry Halpin and Roi Blanco. Machine-learning for spammer detection in crowd-sourcing. In *Workshop on Human Computation at AAAI, Technical Report WS-12-08*, pages 85–86, 2012.

[59] Björn Hartmann and Panagiotis G Ipeirotis. What's the right price? pricing tasks for finishing on time. 2011.

[60] J M Hellerstein and D L Tennenhouse. Searching for jim gray: a technical overview. *Communications of the ACM*, 54(7):77–87, 2011.

[61] Chien-Ju Ho and Jennifer Wortman Vaughan. Online task assignment in crowdsourcing markets. In *AAAI*, 2012.

[62] Jeff Howe. The rise of crowdsourcing. *Wired Magazine*, 14(6):1–4, 2006.

[63] Eric Huang, Haoqi Zhang, David C Parkes, Krzysztof Z Gajos, and Yiling Chen. Toward automatic task design: a progress report. In *Proceedings of the ACM SIGKDD workshop on human computation*, pages 77–85. ACM, 2010.

[64] Shih-Wen Huang. Crowdsummary: Crowdsourced abstractive summary generation with an intelligent interface. *Technical Report. University of Illinois at Urbana-Champaign.*

[65] Nguyen Quoc Viet Hung, Nguyen Thanh Tam, Ngoc Tran Lam, and Karl Aberer. An evaluation of aggregation techniques in crowdsourcing. In *WISE (2)*, pages 1–15, 2013.

[66] Panagiotis G Ipeirotis, Foster Provost, and Jing Wang. Quality management on amazon mechanical turk. In *Proceedings of the ACM SIGKDD workshop on human computation*, pages 64–67, 2010.

[67] Panos Ipeirotis. Demographics of mechanical turk: Now live! (april 2015 edition). `http://www.behind-the-enemy-lines.com/2015/04/demographics-of-mechanical-turk-now.html`, 2015. [Online; accessed 10-May-2016].

[68] David R. Karger, Sewoong Oh, and Devavrat Shah. Budget-optimal task allocation for reliable crowdsourcing systems. *CoRR*, abs/1110.3564, 2011.

[69] Nicolas Kaufmann, Thimo Schulze, and Daniel Veit. More than fun and money. worker motivation in crowdsourcing - A study on mechanical turk. In *AMCIS*, 2011.

[70] Joy Kim and Andres Monroy-Hernandez. Storia: Summarizing social media content based on narrative theory using crowdsourcing. *arXiv preprint arXiv:1509.03026*, 2015.

[71] Aniket Kittur, Ed H Chi, and Bongwon Suh. Crowdsourcing user studies with mechanical turk. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 453–456. ACM, 2008.

[72] Aniket Kittur, Jeffrey V Nickerson, Michael Bernstein, Elizabeth Gerber, Aaron Shaw, John Zimmerman, Matt Lease, and John Horton. The future of crowd work. In *Proceedings of the 2013 conference on Computer supported cooperative work*, pages 1301–1318. ACM, 2013.

[73] Aniket Kittur, Boris Smus, Susheel Khamkar, and Robert E Kraut. Crowdforge: Crowdsourcing complex work. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 43–52. ACM, 2011.

[74] Shailesh Kochhar, Stefano Mazzocchi, and Praveen Paritosh. The anatomy of a large-scale human computation engine. In *Proceedings of the acm sigkdd workshop on human computation*, pages 10–17. ACM, 2010.

[75] Michal Kosinski, Yoram Bachrach, Gjergji Kasneci, Jurgen Van-Gael, and Thore Graepel. Crowd iq: Measuring the intelligence of crowdsourcing platforms. In *Proceedings of the 4th Annual ACM Web Science Conference*, pages 151–160. ACM, 2012.

[76] Pavel Kucherbaev, Florian Daniel, Stefano Tranquillini, and Maurizio Marchese. Crowdsourcing processes: a survey of approaches and opportunities. *IEEE Internet Computing*, 20(2):50–56, 2016.

[77] Pavel Kucherbaev, Florian Daniel, Stefano Tranquillini, and Maurizio Marchese. Relauncher: crowdsourcing micro-tasks runtime controller. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*, pages 1609–1614. ACM, 2016.

[78] Anand Kulkarni, Matthew Can, and Björn Hartmann. Collaboratively crowdsourcing workflows with turkomatic. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*, pages 1003–1012. ACM, 2012.

[79] Anand Kulkarni, Philipp Gutheim, Prayag Narula, David Rolnitzky, Tapan Parikh, and Björn Hartmann. Mobileworks: Designing for quality in a managed crowdsourcing architecture. *Internet Computing, IEEE*, 16(5):28–35, 2012.

[80] Abhimanu Kumar and Matthew Lease. Modeling annotator accuracies for supervised learning. In *Proceedings of the Workshop on Crowdsourcing for Search and Data Mining (CSDM) at the Fourth ACM International Conference on Web Search and Data Mining (WSDM)*, pages 19–22, 2011.

[81] J Richard Landis and Gary G Koch. The measurement of observer agreement for categorical data. *biometrics*, pages 159–174, 1977.

[82] Walter S Lasecki, Kyle I Murray, Samuel White, Robert C Miller, and Jeffrey P Bigham. Real-time crowd control of existing interfaces. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 23–32. ACM, 2011.

[83] Hoong Chuin Lau and Lei Zhang. Task allocation via multi-agent coalition formation: Taxonomy, algorithms and complexity. In *Tools with Artificial Intelligence, 2003. Proceedings. 15th IEEE International Conference on*, pages 346–350. IEEE, 2003.

[84] Edith Law and Luis von Ahn. Human computation. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 5(3):1–121, 2011.

[85] Vincent Leroy, Sihem Amer-Yahia, Eric Gaussier, and Hamid Mirisaee. Building representative composite items. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1421–1430. ACM, 2015.

[86] Guoliang Li, Jiannan Wang, Yudian Zheng, and Michael Franklin. Crowdsourced data management: A survey. *Knowledge and Data Engineering, IEEE Transactions on*, PP(99), 2016.

[87] Christopher H Lin, Ece Kamar, and Eric Horvitz. Signals in the silence: Models of implicit feedback in a recommendation system for crowdsourcing. In *AAAI*, pages 908–915, 2014.

[88] Wang Ling, Luis Marujo, Chris Dyer, Alan Black, and Isabel Trancoso. Crowdsourcing high-quality parallel data extraction from twitter. In *In Proceedings of the Ninth Workshop on Statistical Machine Translation (WMT*. Citeseer, 2014.

[89] Greg Little, Lydia B Chilton, Max Goldman, and Robert C Miller. Exploring iterative and parallel human computation processes. In *Proceedings of the ACM SIGKDD workshop on human computation*, pages 68–76. ACM, 2010.

[90] Greg Little, Lydia B Chilton, Max Goldman, and Robert C Miller. Turkit: human computation algorithms on mechanical turk. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, pages 57–66. ACM, 2010.

[91] Elena Lloret, Laura Plaza, and Ahmet Aker. Analyzing the capabilities of crowdsourcing services for text summarization. *Language resources and evaluation*, 47(2):337–369, 2013.

[92] Christoph Lofi, Joachim Selke, and Wolf-Tilo Balke. Information extraction meets crowdsourcing: A promising couple. *Datenbank-Spektrum*, 12(2):109–120, 2012.

[93] Ngoc-Quang Luong, Laurent Besacier, and Benjamin Lecouteux. Towards accurate predictors of word quality for machine translation: Lessons learned on french–english and english–spanish systems. *Data & Knowledge Engineering*, 96:32–42, 2015.

[94] Kurt Luther, Nathan Hahn, Steven P Dow, and Aniket Kittur. Crowdlines: Supporting synthesis of diverse information sources through crowdsourced outlines. In *Third AAAI Conference on Human Computation and Crowdsourcing*, 2015.

[95] Andrew Mao, Ece Kamar, Yiling Chen, Eric Horvitz, Megan E Schwamb, Chris J Lintott, and Arfon M Smith. Volunteering versus work for pay: Incentives and tradeoffs in crowdsourcing. In *First AAAI conference on human computation and crowdsourcing*, 2013.

[96] Adam Marcus, Aditya Parameswaran, et al. Crowdsourced data management: Industry and academic perspectives. *Foundations and Trends® in Databases*, 6(1-2):1–161, 2015.

[97] Adam Marcus, Eugene Wu, David R Karger, Samuel Madden, and Robert C Miller. Crowdsourced databases: Query processing with people. In *5th Biennial Conference on Innovative Data Systems Research (CIDR'11)*. CIDR, 2011.

[98] David B. Martin, Benjamin V. Hanrahan, Jacki O'Neill, and Neha Gupta. Being a turker. In *CSCW*, pages 224–235, 2014.

[99] Winter Mason and Siddharth Suri. Conducting behavioral research on amazon's mechanical turk. *Behavior research methods*, 44(1):1–23, 2012.

[100] Winter Mason and Duncan J Watts. Financial incentives and the performance of crowds. *ACM SigKDD Explorations Newsletter*, 11(2):100–108, 2010.

[101] Robert R Morris, Mira Dontcheva, and Elizabeth M Gerber. Priming for better performance in microtask crowdsourcing environments. *IEEE Internet Computing*, 16(5):13–19, 2012.

[102] Benedikt Morschheuser, Juho Hamari, and Jonna Koivisto. Gamification in crowdsourcing: a review. In *System Sciences (HICSS), 2016 49th Hawaii International Conference on*, pages 4375–4384. IEEE, 2016.

[103] Thanh Tam Nguyen. Multi-label answer aggregation for crowdsourcing. Technical report, 2016.

[104] David Oleson, Alexander Sorokin, Greg P Laughlin, Vaughn Hester, John Le, and Lukas Biewald. Programmatic gold: Targeted and scalable quality assurance in crowdsourcing. *Human computation*, 11(11), 2011.

[105] Aditya Ganesh Parameswaran, Hyunjung Park, Hector Garcia-Molina, Neoklis Polyzotis, and Jennifer Widom. Deco: declarative crowdsourcing. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 1203–1212. ACM, 2012.

[106] PHP.net. Php: similar_text - manual. `http://php.net/manual/en/function.similar-text.php`, 2016. [Online; accessed 24-Mar-2016].

[107] Julien Pilourdault, Sihem Amer-Yahia, Dongwoon Lee, and Senjuti Basu Roy. Motivation-aware task assignment in crowdsourcing. In *Proceedings of the 20th International Conference on Extending Database Technology*, 2017.

[108] François Portet, Ehud Reiter, Albert Gatt, Jim Hunter, Somayajulu Sripada, Yvonne Freer, and Cindy Sykes. Automatic generation of textual summaries from neonatal intensive care data. *Artificial Intelligence*, 173(7-8):789–816, 2009.

[109] Dragomir Radev, Timothy Allison, Sasha Blair-Goldensohn, John Blitzer, Arda Celebi, Stanko Dimitrov, Elliott Drabek, Ali Hakim, Wai Lam, Danyu Liu, et al. Mead-a platform for multidocument multilingual text summarization. 2004.

[110] Habibur Rahman, Senjuti Basu Roy, Saravanan Thirumuruganathan, Sihem Amer-Yahia, and Gautam Das. Task assignment optimization in collaborative crowdsourcing. In *ICDM*, pages 949–954, 2015.

[111] Peter Robinson. Task complexity, cognitive resources, and syllabus design: A triadic framework for examining task influences on sla. *Cognition and second language instruction*, 288, 2001.

[112] Senjuti Basu Roy, Sihem Amer-Yahia, Ashish Chawla, Gautam Das, and Cong Yu. Constructing and exploring composite items. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, pages 843–854. ACM, 2010.

[113] Senjuti Basu Roy, Ioanna Lykourentzou, Saravanan Thirumuruganathan, Sihem Amer-Yahia, and Gautam Das. Task assignment optimization in knowledge-intensive crowdsourcing. *The VLDB Journal*, 24(4):467–491, 2015.

[114] Jeffrey M Rzeszotarski, Ed Chi, Praveen Paritosh, and Peng Dai. Inserting micro-breaks into crowdsourcing workflows. In *First AAAI Conference on Human Computation and Crowdsourcing*, 2013.

[115] Gerard Salton, Amit Singhal, Mandar Mitra, and Chris Buckley. Automatic text structuring and summarization. *Information Processing & Management*, 33(2):193–207, 1997.

[116] Onn Shehory and Sarit Kraus. Methods for task allocation via agent coalition formation. *Artificial Intelligence*, 101(1):165–200, 1998.

[117] Julius Sim and Chris C Wright. The kappa statistic in reliability studies: use, interpretation, and sample size requirements. *Physical therapy*, 85(3):257–268, 2005.

[118] Elena Simperl. How to use crowdsourcing effectively: Guidelines and examples. *Liber Quarterly*, 25(1), 2015.

[119] Amazon Mechanical Turk. Requester best practices guide. *Amazon Web Services*, 2011.

[120] Chris Van Pelt and Alex Sorokin. Designing a scalable crowdsourcing platform. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, pages 765–766. ACM, 2012.

[121] Luis Von Ahn and Laura Dabbish. Labeling images with a computer game. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 319–326. ACM, 2004.

[122] Jeroen Vuurens, Arjen P de Vries, and Carsten Eickhoff. How much spam can you take? an analysis of crowdsourcing results to increase accuracy. In *Proc. ACM SIGIR Workshop on Crowdsourcing for Information Retrieval (CIR'11)*, pages 21–26, 2011.

[123] Jiannan Wang, Tim Kraska, Michael J Franklin, and Jianhua Feng. Crowder: Crowdsourcing entity resolution. *Proceedings of the VLDB Endowment*, 5(11):1483–1494, 2012.

[124] Fabian L Wauthier and Michael I Jordan. Bayesian bias mitigation for crowdsourcing. In *Advances in neural information processing systems*, pages 1800–1808, 2011.

[125] John White, Theresa O'Connell, and Francis O'Mara. The arpa mt evaluation methodologies: evolution, lessons, and future approaches. In *Proceedings of the 1994 Conference, Association for Machine Translation in the Americas*, pages 193–205, 1994.

[126] Heting Wu, Hailong Sun, Yili Fang, Kefan Hu, Yongqing Xie, Yangqiu Song, and Xudong Liu. Combining machine learning and crowdsourcing for better

understanding commodity reviews. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[127] Rui Yan, Mingkun Gao, Ellie Pavlick, and Chris Callison-Burch. Are two heads better than one? crowdsourced translation via a two-step collaboration of non-professional translators and editors. In *ACL (1)*, pages 1134–1144, 2014.

[128] Jie Yang, Judith Redi, Gianluca DeMartini, and Alessandro Bozzon. Modeling task complexity in crowdsourcing. 2016.

[129] Man-Ching Yuen, Irwin King, and Kwong-Sak Leung. Task recommendation in crowdsourcing systems. In *Proceedings of the first international workshop on crowdsourcing and data mining*, pages 22–26. ACM, 2012.

[130] Omar F Zaidan and Chris Callison-Burch. Crowdsourcing translation: Professional quality from non-professionals. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1220–1229. Association for Computational Linguistics, 2011.

[131] Yudian Zheng, Jiannan Wang, Guoliang Li, Reynold Cheng, and Jianhua Feng. QASCA: A quality-aware task assignment system for crowdsourcing applications. In *SIGMOD*, pages 1031–1046, 2015.